Optimality Theory and Orthography: Using OT to Reconstruct Elamite Phonology

LIN 1290 Forum Paper

Eric J. M. Smith

16-sep-04

Table of Contents

1 Introduction

The main purpose of this paper is to consider the problems involved in applying techniques from Optimality Theory to the study of orthography. As an exercise of these techniques, a suitably challenging problem was chosen, namely the reconstruction of the phonology of Elamite, a language which is known only from written sources.

The impetus for this project was the claim that Optimality Theory is a learning algorithm. Presented with only a set of overt forms, a language learner is supposed to be able to derive both the underlying forms and the set of OT constraints. If the mechanisms provided by OT are powerful enough for a language learner to acquire a natural language, it should be possible to apply the same mechanisms to the process of "learning" an unknown language given only its orthography. In this application of OT, the overt form of any given word would be its orthography while the underlying form would be its phonology. Although the nature of the constraints will be slightly different from what is typical of OT, it should still be possible to rank those constraints and then to use OT's Lexicon Optimization module to derive the underlying forms.

The particular application of these techniques to Elamite was motivated largely because Elamite phonology is currently so poorly understood. Given the language's lack of known affiliations, spelling irregularities provide one of the key pieces of evidence to give us insight into the underlying phonology. If the techniques being proposed in this paper do prove to have merit, the hope is that, as a side effect of evaluating this new application of Optimality Theory, we might also find some answers to questions about Elamite phonology which had previously been unanswerable.

The choice of Elamite did introduce some additional challenges. By choosing such a poorly understood language, we give up the possibility of corroborating our results against other phonological evidence. In addition, the cuneiform writing system is a complex one, and introduces a number of complications

which are at best tangential to the main problem of applying OT techniques to orthographic analysis.

The paper will start by giving a description of several proposed reconstructions of Elamite phonology. This will be followed by a brief overview of the theory of writing systems. Then the attention turns to Optimality Theory, starting with a general discussion of how to apply Optimality Theory to orthographic data, followed by specific details of how the major OT modules (GEN, H-EVAL, and Lexicon Optimization) were implemented for this project. The paper will conclude with a discussion of the results obtained by applying the learning algorithm to the Elamite data.

1.1 A note on transcription

The transcription of cuneiform texts has a long history, which includes standards for transcription which are different from those which would typically be used by linguists. In particular, this means that sequences of graphemes will be transcribed in the traditional style used in cuneiform studies, with graphemes separated by hyphens and words separated by periods. So for example, *da-iš* will be written rather than the more cumbersome <da>+<iʃ>.

Conventions from cuneiform studies which may be unfamiliar to mainstream linguists include the use of accents and subscripts to indicate different graphemes which are read with the same phonetic value. So for instance *ša* (), *šá* (), and *šà* () are three graphemes which are all read (in Akkadian at least) with the value /ʃa/, while tu_4 () is the fourth grapheme with a value of /tu/. In addition, the practice in this field is to use non-IPA transcriptions (e.g. /y/ for IPA /j/, /ǰ/ for IPA /dʒ/, /č/ for /tʃ/, and so on).

Certain special graphemes, known as determinatives, are attached to nouns to indicate the class of item being referred to. This practice is largely a holdover from Sumerian, which had a much richer set of determinatives. In transcription, these determinatives are typically written as subscripts, but we will follow the practice of Hinz & Koch (1987), which is to use a period to separate the

determinative from the word being modified. The commonest determinatives in Elamite are *h* (a geographical location), *hh* (a male human, used during the Achæmenid period), *hw* (a male human, used in the Neo-Elamite period), *v* (a male human, following Akkadian usage), *f* (a female human), and *d* (a divinity). So something like *v.hal.lu-tu-uš.d.in-su-uš-na-ak* 'the land enriched (the God) Inšušinak?' would be the personal name of a male human which happens to include a theophoric element.

1.2 A note on software

The bulk of the research described here was carried out with the aid of a piece of software specifically developed for this project. The software is nicknamed *Grotefend*, after Georg Friedrich Grotefend, who was responsible for the preliminary decipherment of the cuneiform inscriptions at Persepolis. The *Grotefend* software was written in C++ using Trolltech's Qt toolkit, and runs under Mac OS X.

The bulk of the data being used by *Grotefend* is stored in a large XML (eXtensible Markup Language) file which contains an entry for each of the dictionary entries in Hinz & Koch (1987). Each entry is tagged with attributes such as morphology, cognates, semantics, and chronology.

While the majority of the code in *Grotefend* is new and written specifically for this study, the portions which implement the Gradual Learning Algorithm (§4.4) were adapted from Paul Boersma's Visual Basic source code from the OTSoft program, which was kindly provided by Bruce Hayes.

2 Elamite Language

2.1 Historical & geographical context

Elamite is an extinct language which was spoken in what is now southwestern and central Iran. The language has no known linguistic affiliations, although a connection to the Dravidian family has been proposed by McAlpin (McAlpin,

1982) and others. McAlpin's hypothesis has failed to find much enthusiasm from either Dravidianists or most students of Elamite.

We have texts from Elam dating back to around 3200 BC, which makes the Elamites second only to the Sumerians in the use of writing. Although the Elamites had two indigenous writing systems, the bulk of the available Elamite-language texts are written in an adapted version of the cuneiform system, which was probably borrowed from the Sumerians via Akkadian. Elamite texts written in cuneiform are attested from around 2400 BCE until roughly 360 BCE.

The historical record of Elam is broken by long gaps, during which the Elamites seem to have vanished into the hills, only to return a few centuries later. These gaps are useful for dividing the textual record into discrete periods. Since the earliest Elamite texts are written in a script which has not been deciphered, we really only have information about the Elamite language starting with their adoption of cuneiform. The Old Elamite language is recorded from about 2400 BCE to 1700 BCE, Middle Elamite from 1300 BCE to 1100 BCE, and Neo-Elamite from 743 BCE to 654 BCE. After the fall of Elam, Elamite was used as an administrative language in the Persian Empire, and this variant of the language is known as Achæmenid Elamite, and is attested from 539 BCE to 360 BCE.

Since both the language and the orthographic practices are certain to have changed over such a long time-span, this study will restrict itself to text from a single era. The Achæmenid Elamite period was chosen, both because this period contains the largest volume of texts, but also because those texts are particularly rich in Old Persian names and loanwords which provide a useful starting point for estimating the phonology. While the *Grotefend* database does contain entries from all periods, processing of entries from the pre-Achæmenid periods is turned off.

2.2 The cuneiform writing system

As originally developed by the Sumerians, the cuneiform writing system was primarily logographic. With the adaptation of cuneiform for writing Akkadian,

greater use was made of phonetic grapheme values. The Elamites took this a step further, and moved to an almost completely phonetic system, with only a handful of logograms.

Within the cuneiform system, the signs which represent phonetic values can be classified as what Sproat (2000) calls a "core syllabary". That is, each grapheme represents a syllable, but the system does not have enough graphemes to represent all of the language's syllables. This is particularly the case for *CVC* graphemes, for which only a fraction of the CVC syllables occurring in the language actually have a corresponding *CVC* grapheme. Such syllables must be written using "syllable telescoping", where a *CV-VC* combination is written, with the internal vowel being repeated. So for example, if there were no *lan* grapheme, the syllable /lan/ would have to be written *la-an*. In fact, even when the *CVC* grapheme does exist, the *CV-VC* writing is often preferred. This is especially the case in Elamite, where the inventory of *CVC* graphemes is greatly reduced compared to Akkadian.

The cuneiform system also poses problems for the phonologist since the phonetic values of many signs are often underspecified. For instance, the *it* grapheme () also has Akkadian values of *id*, *et*, and *ed*. In order to represent the grapheme's phonology in featural terms, we would have to say that the vowel it represents is [+front] and [-low] but underspecified for [high]. Similarly, we would have to say that the consonant portion of the grapheme's value is underspecified for [voice].

2.2.1 Broken writing

Elamite makes use of a variation on telescopic writing not found in Akkadian, wherein certain syllables are written using a CV_1-V_2C sequence (V_1 being different from V_2). The meaning of this "broken writing" is unclear, but a number of possibilities have been suggested:

1) V_1 and V_2 are articulated separately as the nuclei of two separate syllables, possibly separated by a glottal stop. So *da-iš* might represent

/da.iʃ/ or /daʔiʃ/. A similar usage occurs occasionally in Hittite cuneiform (Gragg, 1996), (Paper, 1955).

2) The combination of V_1 and V_2 is simply a diphthong. In favour of this, it turns out that the value of V_2 is always either *i* or *u* (Paper, 1955).

3) The combination of V_1 and V_2 is an attempt to represent a vowel of some intermediate quality that could not otherwise be represented within the cuneiform system. So *za-um* might represent /zom/ (Hinz and Koch, 1987) and *da-iš* might represent /dɛʃ/ (Paper, 1955)

4) The value of V_2 is simply ignored, and the second sign is simply being used to provide the consonant that closes the syllable. Under this interpretation, *da-iš* is just a way of writing/daʃ/ (Paper, 1955)

One other logical possibility, namely that the value of V_1 is ignored and the broken vowel sequence represents V_2, does not seem to have been proposed by anyone studying the language.

2.3 Previous studies of Elamite

Although Elamite, Akkadian, and Old Persian inscriptions were discovered at the same time in the 19th century, the study of Elamite has lagged behind that of the other two languages. The key figure in early Elamite studies was F. H. Weißbach, who was active in the first decades of the 20th century. Since that time, the study of Elamite can be grouped into two schools. The French, whose excavations were centred at Susa, the Elamite capital in southwestern Iran, concentrated on inscriptions and documents from the Old Elamite, Middle Elamite, and Neo-Elamite periods. The so-called Chicago school, largely working further east at Persian sites such as Persepolis and Pasargadae, focused more on the Achæmenid Elamite period.

The following section will give a brief overview of the major works in the field of Elamite phonology, concentrating primarily on the reconstruction of the

inventory, but also with discussion of the various theories proposed to explain the orthographic oddities of Elamite.

2.3.1 *The Phonology and Morphology of Royal Achæmenid Elamite* (Paper, 1955)

H. H. Paper was one of the founding fathers of the Chicago school of Elamite studies. His very short book is restricted to a narrowly defined body of texts, namely the royal inscriptions of the Achæmenid period. These texts lend themselves well to analysis since they tend to be bilinguals or trilinguals, and since they contain numerous Old Persian personal and geographic names. These two factors make the Achæmenid royal inscriptions a particularly rich source of information about Elamite phonology.

Paper's approach to reconstruction centres on the comparison of Elamite orthographies with known Old Persian pronunciations. He tends to be conservative in his reconstructions, and his inventory omits several phonemes which subsequent authors feel to be justified. However, it is worth describing his conclusions in some detail, since his work provides the basis upon which subsequent scholars were able to build.

Paper concludes that there is no voicing distinction in Elamite, since the same set of signs is used for both Old Persian /b/, /p/, and possibly /f/. The same indifference to voicing is shown in the orthography for the set of Old Persian phonemes/d/, /t/, /θ/ and for the set /g/, /k/, /x/.

By observing the variant spellings of sibilants in loanwords, Paper comes up with a series of three sibilants. The first two are /s/ and /š/, which are clearly distinct, but whose exact phonetic values are uncertain. The third one corresponds to Old Persian /č/, and is written using graphemes containing the Akkadian value /ṣ/ (e.g. *ṣa*, *ṣi*, *aṣ*).

Paper's reconstruction of the vowel system is conservative, and he considers only the basic set of /a/, /i/, and /u/ to be distinct. Although there are certain words which are consistently written using *Ce* graphemes, Paper does not

consider that to be sufficient evidence for an /e/ phoneme. He also does not believe there to be any evidence to justify the reconstruction of an /o/ phoneme.

In terms of interpreting the writing system, Paper notes that there are some geminate spellings, but considers them to be merely an orthographic convention, and without phonetic significance. His position on broken writings corresponds to the fourth one listed in §2.2.1 namely that CV_1-V_2C writings ignore the value of V_2 and represent CV_1C. That being said, he does admit the possibility of /ai/ and /au/ diphthongs in Old Persian loanwords, as well as in two native Elamite words.

Paper proposes a /y/ semivowel, although it is represented in the texts only in the grapheme *ya*. He does not believe that there is sufficient evidence to support the existence of a /w/. Supporting this, he notes that Old Persian names containing /w/ tend to be written using an *mV* grapheme.

The value of the various *hV* graphemes is problematic in Achæmenid Elamite, since *hV* graphemes often seem to be in alternation with the corresponding *V* grapheme, as in *hi-da-ka*$_4$ vs. *i-da-ka*$_4$ 'with'. Based on this, Paper concludes that these *hV* graphemes are an orthographic convention and there is insufficient evidence to support the existence of an /h/ phoneme. Other authors, such as Grillot-Susini (1988) have suggested that earlier dialects of Elamite did have an /h/ phoneme, but it was not part of the dialect which appears in the Achæmenid period texts. Whatever the case, *hV* graphemes are clearly not being used to write an /x/ phoneme (as was the case in Akkadian), since Old Persian /x/ corresponds to a *kV* or *Vk* grapheme when it is written in Elamite.

Paper also points out a couple of mechanisms where he believes that a superfluous vowel is being written to represent a consonant cluster which would otherwise not be representable in cuneiform. The first mechanism involves writing a prothetic vowel, so for example the initial cluster in *Skudra* 'Thracian' is written *iš-ku-ud-ra*, while the final cluster in *dušt* 'who took' is written as either *du-iš-ti* or *du-iš-da*. The second mechanism involves the use of a *CVC* grapheme

where the vowel's value is ignored, as in the writing of the Old Persian personal name *Fra□da* as *pír-ra-da*.

(1) Paper's inventory of phonemes

p	*t*		*k*	*i*		*u*
	s	*š*			*e*	
		č				*a*
m	*n*					
l		*r*				
		y				

2.3.2 *The Elamite Language* (Reiner, 1969)

Reiner's book-length article was originally written in 1960, but was not published until 1969. It represents a continuation of Paper's work, extending it to cover other dialects of Elamite besides Royal Achæmenid.

Like Paper, she considers broken CV_1-V_2C writings to represent CV_1C, and considers this to be verging upon alphabetic use. In effect, the V_2C grapheme is being used purely as a consonant. She bolsters this position with an analysis of the reduced inventory of *VC* graphemes in Achæmenid times. It is interesting however, that in the other use of graphemes purely for their consonantal value, namely the representation of word-final consonant clusters, it is invariably a *CV* grapheme which is employed, and not one of Reiner's proposed *VC* pseudo-consonants.

Reiner's chief contribution to the reconstruction effort is her observation that there is a pattern in the use of geminate spellings for Old Persian and Akkadian loanwords. Intervocalically, a voiceless stop in Old Persian or Akkadian is always rendered as a geminate consonant in the Elamite orthography, while an intervocalic voiced stop is written as a non-geminate.

Reiner points to a parallel orthographic strategy from Tamil (which has no voicing distinction), where intervocalic voiceless consonants in Sanskrit loanwords are written as geminates. Within languages using the cuneiform

system, a similar use of geminate consonants to indicate voicelessness has also been proposed for Hittite and Hurrian.

While Reiner's rule does hold in general, there are a number of exceptions. She points out a couple herself, and more counterexamples have been found since the publication of her article (Hinz and Koch, 1987).

2.3.3 *Proto-Elamo-Dravidian: The Evidence and its Implications* (McAlpin, 1982)

McAlpin comes to the study of Elamite from a different direction than Paper or Reiner, being a Dravidianist rather than an Assyriologist. However, he does use Paper's reconstruction as a starting point, adding modifications that he feels are justified by his comparative studies.

McAlpin reconstructs two series of stops, one being lax, commonly voiced, and written with a single consonant, and the other being tense, commonly voiceless, and written with a geminate. For McAlpin this tense/lax distinction is not restricted to loanwords, but extends to native Elamite words as well.

Elamite also has some words which consistently employ geminate spellings of /l/ and /r/ (e.g. *tal-lu* 'to write', never appears as **ta-lu*). In this case, McAlpin suggests that the geminate spelling might represent retroflexion, and the non-geminate spelling represents an alveolar articulation. His choice of retroflexion as a distinctive feature is probably motivated by the presence of retroflex liquids in Dravidian languages.

Differing from Paper, he reconstructs Paper's /č/ phoneme as /z/, but concedes that the loanword data suggests it may actually have been phonetically [tʃ]. According to McAlpin, both Elamite /s/ and /z/ derive ultimately from Proto-Elamo-Dravidian /*c/. McAlpin does not further explain the phonetic details of this reconstructed /*c/, but it corresponds to /s/ and /z/ in Elamite and to /tʃ/, /s/, or /k/ in modern Dravidian languages.

McAlpin also diverges from Paper in a couple of minor respects. He proposes an /e/ vowel, but suggests that it is only distinguished from /i/ when in a word-initial syllable. McAlpin believes stress to be on the initial syllable, and he attributes the numerous *e*/*i*/*u* alternations in word-final position to the presence of an underspecified unstressed vowel, which he refers to as /ə/. In this, he differs from Paper and Reiner, who take these alternations as evidence that there was no final vowel, but merely an orthographic convention for writing word-final consonant clusters.

McAlpin is also the first to propose nasalized vowels in alternating spellings such as *hi-in-du-iš* vs. *hi-du-iš* 'India' and *hu-um-ba-an* vs. *hu-ba-an* '(the god) Humban'.

McAlpin also proposes a /w/ phoneme, which is indicated in the orthography by an otherwise superfluous *ú*, as in the spelling *da-a-hi-ú-uk-ka*$_4$ for the Old Persian name *Dahiwukka*. However, the evidence for this is weak, since there are numerous occurrences of *ú* graphemes which do not appear to correspond to an underlying /w/.

2.3.4 *Eléments de grammaire élamite* (Grillot-Susini and Roche, 1988)

Grillot-Susini's work serves as a good summary of the French school of Elamite-language studies. Her primary interest is in the syntax and morphology of Elamite, but she does devote a brief section to the language's phonology.

Unlike the Chicago school authors, Grillot-Susini believes that the language does have a voicing distinction, present in both stops and sibilants. The evidence that has generally been used to show the lack of voicing distinction consists of the loanword data cited by Paper, as well as spelling alternations such *ba-ri-iš-da* vs. *pa-ri-iš-da* 'they went'. While Grillot-Susini does admit that this kind of variation occurs for some words, she points to numerous other words that are consistent in their use of graphemes and never display such variation. In particular, she claims to have found at least one minimal pair, *ki-ri* 'goddess' vs. *gi-ri* 'gratitude'.

Grillot-Susini notes the same alternations of nasals that McAlpin used as evidence for nasal vowels. However, instead of reaching the same conclusion as McAlpin, she proposes underlying nasal segments and a set of phonological rules which eliminate the nasals in certain contexts, namely $n \rightarrow m \,/\, \_p$ and $m \rightarrow \emptyset \,/\, \_p$. This explains the same phenomena, but seems less elegant than McAlpin's proposal.

(2) Grillot-Susini's inventory of phonemes

p		*t*		*k*		*i*			*u*
b		*d*		*g*			*e*		
		s	*š*					*a*	
		z							
					h				
m		*n*							
	l		*r*						

Grillot-Susini does not explicitly state a preference for any one of the four interpretations of broken writings, but her transcriptions generally follow the fourth (CV_1-V_2C ← /CV_1C/) interpretation (e.g. *ha-li-en-ka*$_4$ ← *halinka*). Her transcriptions do include occasional diphthongs which correspond to the presence of an extra *ú* grapheme in the orthography (e.g. *šá-ú-mar-ráš* ← *šaumarraš*).

Her reading of the texts does differ in one significant regard from earlier authors, in that she maintains that geminate spellings really do represent geminate pronunciations, so a spelling like *mu-uš-šá* (or even *mu-uh-šá*) would be read as *mušša*. However, the opposite relation is not true, in that a non-geminate spelling may still conceal a geminate pronunciation.

2.3.5 *The Elamite Language* (Khachikjan, 1998)

Margaret Khachikjan comes from what Stolper (2001) describes as a third tradition of Elamite studies, separate from the French and Chicago schools. Khachikjan was a student of I. M. Diakonoff, who was one of the earliest proponents of a connection between Elamite and Dravidian, so it is not surprising that Khachikjan is one of the few authors to actively endorse McAlpin's Proto-Elamo-Dravidian hypothesis.

Her reconstruction follows McAlpin in most respects, including his tense/lax distinction for stops, and the use of *ll* and *rr* spellings to indicate retroflexion. The main departure from McAlpin is that she argues against the existence of semi-vowels /y/ and /w/, and that she proposes the existence of a bilabial fricative of some sort.

Khachikjan does pay particular attention to spelling alternations when attempting to recreate the phonetic reconstructions of the various phonemes. So for instance, based on the existence of an alternation *te-em-ti/si-im-ti/še-em-ti* in the spelling of the word meaning 'lord', she concludes that the initial segment must be an affricate, which she reconstructs as /c/. Based on her argumentation, it appears that her /c/ corresponds to IPA /ts/.

(3) Khachikjan's inventory of phonemes

p	*t*		*k*	*i*		*u*
p′	*t′*		*k′*	*e*		
	s	*š*			*a*	
	c	*č*				
v/f			*h*			
m	*n*					
l		*r*				
ll		*rr*				

2.4 Source data

The data being used for this study is taken largely from the *Elamisches Wörterbuch* (Hinz and Koch, 1987). Earlier works such as Hallock (1969) and König (1965) have provided glossaries of Elamite words for specific corpora, but the *Wörterbuch* is the only source which incorporates Elamite data from all historical periods. It has the virtue of containing every single attested form known to the authors. This range of forms is particularly useful for this project, since we have special interest in alternate spellings of given words.

For the most part Hinz and Koch do not advance their own theories about Elamite phonology. Instead, they content themselves with presenting the available data, together with comments and theories of previous scholars.

There is however one case where Hinz and Koch do put forward an interesting interpretation of the orthography. Sequences like *ik-ba* and *uk-ba* are relatively common in Achæmenid Elamite. They suggest that the combination of the voiceless velar and the voiced bilabial are intended to indicate a voiceless bilabial. So *hh.uk-ba-kar-na* would be the Elamite spelling for the Old Persian name *Upakrna*.

As part of the preparation of this text, the pages of the *Elamisches Wörterbuch* were scanned as JPEG images, in the hope that they could then be processed using Optical Character Recognition software. However, initial attempts at using OCR on the dictionary pages proved to be a failure. The use of unusual diacritics, the mixture of languages, and the layout of the dictionary entries were all too much for the OCR software (OmniPage Pro) which is really intended for recognising blocks of words in a single language. Since the OCR approach was not viable, the text had to be entered manually. Even if the OCR approach had been successful, an extra stage of manual processing would still have been necessary in order to regularise the data into a format that could be manipulated by the *Grotefend* software.

2.5 Hypotheses to be tested

The strategy of this paper is to use the techniques of Optimality Theory to reconstruct the language's phonology. Rather than start from scratch, we will instead take the various hypotheses presented by earlier authors and attempt to encapsulate each of those hypotheses in the form of an OT constraint. By applying these constraints, it should then be possible to reconstruct the underlying phonology of the language.

For the sake of clarity, the hypotheses to be tested will be given labels so that they can be referred to easily. Within each major grouping (e.g. H1), the sub-hypotheses (e.g. H1a, H1b, H1c, and H1d) all refer to a related context or a related orthographic phenomenon. The importance of these groupings will become clear when we discuss GEN (§5.3) and Lexicon Optimization (§7). Within each group, individual sub-hypotheses may or may not be mutually exclusive. The hypotheses to be tested are as follows:

H1) Interpretation of broken *CV*$_1$-*V*$_2$*C* writings (see §2.2.1 for details)

- H1a) The written vowels of the *V*$_1$-*V*$_2$ sequence are articulated as two separate spoken vowels, possibly separated by a glottal stop (e.g. /daʔiʃ/ → *da-iš*).
- H1b) The *V*$_1$-*V*$_2$ sequence is being used to represent a diphthong (e.g. /dajʃ/ → *da-iš*).
- H1c) The combination of *V*$_1$ and *V*$_2$ in the orthography is being used to represent an intermediate vowel which could not otherwise be written in cuneiform (e.g. /dɛʃ/ → *da-iš*).
- H1d) The *V*$_1$-*V*$_2$ sequence in the orthography is simply being used to indicate an underlying phonology of *V*$_1$; the presence of *V*$_2$ in the orthography is merely a scribal convention (e.g. /daʃ/ → *da-iš*).

H2) Voicing of stops

H2a) The language's phonology includes a true voicing distinction, and this distinction is reflected in the choice of graphemes with voiced or unvoiced values (Grillot-Susini and Roche, 1988).

H2b) The choice of graphemes with voiced or voiceless values is significant, but the opposition being represented is tense/lax, or some other distinction than voicing. It seems likely that there may not be enough information to distinguish this from H2a.

H2c) The choice of graphemes using voiced and voiceless grapheme values does not reflect a distinction in the phonology. The choice of graphemes is merely an orthographic convention.

H2d) Voicelessness is indicated using the orthographic mechanism suggested by Hinz & Koch (1987), which was described in described in §2.4. The voicing feature is supplied by one grapheme and the place of articulation by another (e.g. /upa/ → *uk-ba*).

H3) Geminate consonants

H3a) Geminate orthographies represent underlying geminate phonologies.

H3b) Geminate orthographies are being used to indicate voicelessness, as suggested by Reiner (1969).

H3c) Certain geminate spellings are used to indicate a distinction other than voicing, such as retroflex/alveolar.

H4) Nasal vowels

H4a) The observed alternations in the writing of nasals indicate the presence of nasal vowels (e.g. /hũban/ → *hu-um-ban, hu-ban*).

H4b) The observed alternations in the writing of nasals can be explained by underlying nasal consonants which are deleted through some

phonological process (e.g. /humban/ → [huban] → *hu-um-ban, hu-ban*).

H5) Word-final vowels

H5a) The alternations in the writing of word-final vowels indicate an attempt to render a word-final consonant cluster which could not otherwise be written in cuneiform.

H5b) The alternations in the writing of word-final vowels indicate the presence of a /ə/ or other underspecified vowel.

H5c) The alternations in the writing of word-final vowels indicate an actual alternation in the phonological vowel.

H6) Sibilants

H6a) The sibilant inventory includes a /č/ (= IPA /tʃ/) which is written using the Akkadian *ṣV* and *Vṣ* graphemes (Paper, 1955).

H6b) The sibilant inventory includes a /z/ which is written using the Akkadian *ṣV* and *Vṣ* graphemes (Grillot-Susini and Roche, 1988).

H6c) The sibilant inventory includes a /c/ (= IPA /ts/) which is written using the *sV*, *šV*, *tV*, *Vs*, *Vš*, or *Vt* graphemes (Khachikjan, 1998).

H7) The phonemic inventory includes an /h/.

H7a) The *hV* and *Vh* graphemes are being used to write the phoneme /h/.

H7b) The *hV* and *Vh* graphemes are purely orthographic variants of the equivalent *V* graphemes (Paper, 1955).

H8) The phonemic inventory includes an /f/ or a /v/.

H8a) The *pír* grapheme is being used to indicate a /fr/ or /vr/ sequence (Khachikjan, 1998).

H8b) The *pír* grapheme is being used to indicate an ordinary /pr/ or /pir/ sequence.

H9) The phonemic inventory includes a /j/, written with the *ya* grapheme.

H9a) The *ya* grapheme is being used to write the phoneme /j/.

H9b) The *ya* grapheme is being used to write a non-syllabic allophone of /i/.

H10) The phonemic inventory includes a /w/, written with the *ú* grapheme.

H10a) The grapheme *ú* is being used to indicate a /w/ (McAlpin, 1982).

H10b) The grapheme *ú* is being used to indicate a /u/.

H11) The phonemic inventory includes an /e/.

H11a) There is an /e/ vowel, distinct from /i/.

H11b) There is an /e/ vowel, but it is distinct from /i/ only in the first syllable of a word (McAlpin, 1982).

H11c) The *e, eC* and *Ce* graphemes are purely orthographic variants of the equivalent *i, iC,* and *Ci* graphemes (Paper, 1955).

There is of course an implicit hypothesis H0, namely that OT can actually be used to shed light on these questions. For several of the above hypotheses, it seems quite possible that no combination of constraints would be able to tease out the desired evidence. The details of the implementation of these hypothesis-specific OT constraints will be the subject of §6.3. Since we are exploring the relationship between phonology and orthography, it would be useful to have a brief theoretical discursion into the nature of that relationship.

3 Theory of Writing Systems

The discussion of the Elamite writing system which is being presented here will be framed within the theory of writing systems proposed by Sproat (2000). Large parts of his theory are not relevant to the questions being explored here, but where appropriate we will make use of his theoretical framework. The core of Sproat's theory is that "particular (sets of) linguistic elements *license* the occurrence of (sets of) orthographic elements". The exact details of which linguistic elements license which orthographic ones are specific to any particular combination of spoken language and writing system. The rest of this section will show how this notion of licensing would apply to Elamite.

3.1 Orthographically Relevant Level

One of the concepts which is central to Sproat's theory is that of Orthographically Relevant Level (ORL). The term "level" in this context is defined by the successive application of phonological rules. The example Sproat uses when describing ORL is the comparison between Belarusian and Russian orthographies. Both languages have a well-known phonological process by which unstressed vowels are reduced. Belarusian orthography reflects the surface phonology, including the application of vowel reduction. Russian orthography represents the underlying phonology without the application of the vowel reduction process. Belarusian is said to be "shallow", while Russian is (comparatively-speaking) "deep".

A good example of a language with a deep orthography is English. The classic demonstration of this is the pair of words *electric* and *electricity*, where the second <c> corresponds to either /k/ or /s/. In this case, rather than reflecting the surface phonology, the <c> in the orthography serves to maintain the identity between different instances of the stem morpheme.

One of the two fundamental claims of Sproat's theory is that of Consistency, which he states as follows:

(4) Consistency (Sproat, 2000)

The ORL for a given writing system (as used for a particular language) represents a consistent level of linguistic representation.

In the context of Elamite orthography, what would be the ORL? We have no direct evidence of the phonological rules that are applied within spoken Elamite, so we must use secondary evidence to determine the orthographic depth.

The first piece of evidence is the fact that Akkadian has a shallow orthography (in Sproat's terminology). When written phonetically, Akkadian orthography reflects the phonology after the application of processes such as nasalization, vowel harmony, and compensatory lengthening. Since the scribal tradition in Elam was heavily influenced by Akkadian models, and since Elamite scribes seem likely to have been bilingual in Akkadian (Steve, 1992), it seems reasonable to assume that they would have retained the same orthographic depth when adapting the Akkadian system to their own language. That being said, we have seen how languages as similar as Russian and Belarusian can have different Orthographically Relevant Levels, so it is quite possible that Elamite had a completely different Orthographically Relevant Level from Akkadian.

A stronger piece of evidence would be that morphemes change their spelling when they are affixed together. If Elamite orthography behaves like our English example of *electric* and *electricity*, preserving the identity of morphemes in the presence of affixes, then we can conclude that the ORL is relatively deep. However, if the spelling of a given morpheme is seen to vary due to the presence of affixes, this is evidence that the ORL is shallow.

As it turns out, there is plenty of evidence for the sorts of morpheme alternations expected from a shallow ORL. The orthography reflects the application of phonological processes such as place assimilation, nasal assimilation, and cluster simplification. An example of how the orthography reflects each of these phonological processes is given in (5).

(5) Evidence of shallow ORL

Stem	Affix	Affixed form
ki-it-ti-in 'length'	*-ma* (locative case)	*ki-it-ti-im-ma* 'in length'
du-ni-ih 'I gave'	*in-* (3Sg object)	*id-du-ni-ih* 'I gave it'
ti-ri-man 'say (imperfect)'	*-k* (past participle?)	*ti-ri-ma-ak* 'it was said'

In Sproat's framework, the ORL is the input for a mapping function, $M_{ORL \rightarrow \Gamma}$, the output of which is the orthography (Γ). This mapping function is itself made up of the concatenation of two mapping functions.

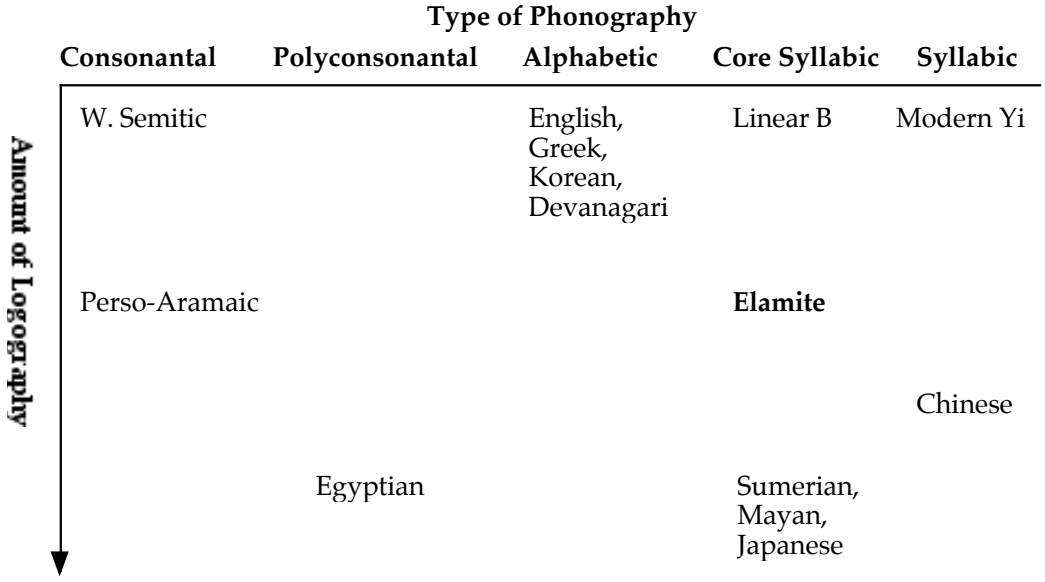
The first function is M_{Encode}, which is a set of graphic encoding rules, taking the units of the ORL as an input and producing that portion of the orthography which can be determined from the phonology. This is the mapping which is the primary interest of this study.

The second function is M_{Spell}, which is a set of autonomous spelling rules, taking the output of M_{Encode} and turning it into correctly formed orthography. In the case of Elamite, we can give M_{Spell} the responsibility for generating those portions of the orthography which are not derived from the phonology, namely graphemes such determinatives.

3.2 Units of encoding

On examining a writing system, one has to consider what size of linguistic elements are being encoded. Numerous taxonomies have been proposed, but the one Sproat comes up with is shown in (6). He calls it "nonarboreal", to contrast it with traditional tree-like taxonomies such as those of Sampson and DeFrancis. In this scheme, the ORL can be thought of as a third dimension, with writing systems being arranged along that dimension according to their orthographic depth.

(6) A nonarboreal classification of writing systems (Sproat, 2000)



As discussed in §2.2, under Sproat's scheme, cuneiform is classified as a "core syllabary", by which he means that the graphemes represent syllables, but only some syllables (typically the open ones) have a corresponding grapheme. Sproat only discusses Sumerian, but Elamite clearly falls into the same "core syllabary" category. Elamite does differ significantly from Sumerian in having a greatly diminished use of logograms, which is indicated by its higher position in (6).

However, it is a bit of a simplification to think of cuneiform as being strictly syllabic in nature. Consider the Elamite orthography of the Old Persian name *Gauma□ta*, which appears as *v.kam-ma-ad-da* or *hh.kam-ma-da*. We can be very secure in our reconstruction of the Old Persian phonology, since the name is attested in Greek and Roman sources, as well as in later Persian-language texts.

We do not know how *Gauma□ta* might have been adapted into Elamite. However, unless we assume that Elamite geminated the /m/ when it adapted the name from Old Persian, or we assume that Elamite had bizarre syllabification rules, a syllabification like /gau.ma.ta/ seems like the most reasonable one (disregarding for now the exact values of the phonemes). If /gau.ma.ta/ is the actual syllabification, then what type of element is represented by the *kam* grapheme? It would appear that *kam* is representing both the entire first syllable

as well the onset of the second one. Similarly, the *ad* grapheme in *v.kam-ma-ad-da* would appear to represent both the rime of one syllable and the onset of another.

By contrast, in Akkadian cuneiform, a geminate orthography will occur only if there is gemination in the underlying phonology (Caplice and Snell, 1988). That is, if *kam-ma-ad-da* were found in an Akkadian text, it would have to correspond to something like /kam.ma.ad.da/. Among languages using cuneiform, this sort of mismatch between the cuneiform orthography and the syllable structure seems to be peculiar to Elamite. The mismatch will cause some minor complications when trying to represent the licensing relationship.

3.3 Representations

Sproat (2000) suggests two representations which are suitable for describing the mapping between orthography and the underlying non-orthographic information. The first of these is an Attribute Value Matrix, as shown in (7), which is an adaptation of the structures used by in linguistic traditions such as Head-driven Phrase Structure Grammar. In this notation, licensing is indicated by numeric coindexation: the grapheme with subscript *i* is licensed by the element with subscript *i**.

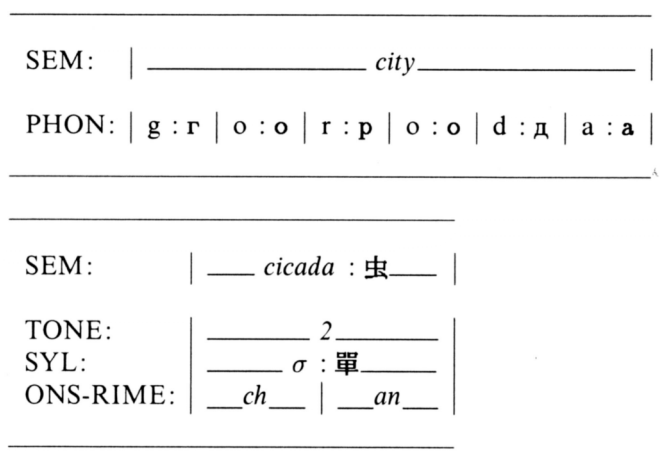
(7) Sample Attribute Value Matrices for Russian and Chinese (Sproat, 2000)

$$\begin{bmatrix} \text{PHON} & \langle g_{1*}\, {}'o_{2*}\, r_{3*}\, o_{4*}\, d_{5*}\, a_{6*} \rangle \\ \text{ORTH} & \{\text{г}_1, \text{о}_2, \text{р}_3, \text{о}_4, \text{д}_5, \text{а}_6\} \\ \text{SYNSEM} & \begin{bmatrix} \text{CAT} & \textit{noun} \\ \text{GEN} & \textit{masc} \\ \text{CASE} & \textit{gen} \\ \text{NUM} & \textit{sing} \\ \text{SEM} & \textit{city} \end{bmatrix} \end{bmatrix}$$

$$\begin{bmatrix} \text{PHON} & \left[\text{SYL} \left[\begin{matrix} \text{SEG} & \left\langle \left[\text{ONS } \textit{ch}\right] \left[\text{RIME } \textit{an}\right] \right\rangle \\ \text{TONE} & 2 \end{matrix} \right] \right]_{1*} \\ \text{SYNSEM} & \begin{bmatrix} \text{CAT} & \textit{noun} \\ \text{SEM} & \textit{cicada}_{2*} \end{bmatrix} \\ \text{ORTH} & \{\text{虫}_2, \text{單}_1\} \end{bmatrix}$$

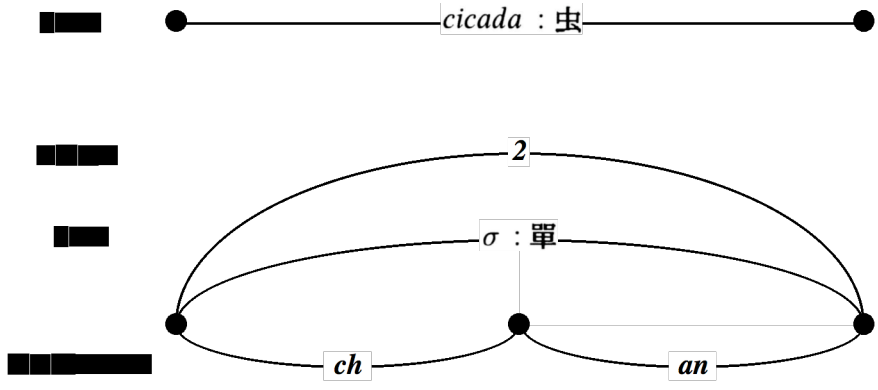
An alternative representation is what Sproat calls an Annotation Graph, based on Bird & Liberman (1999). Two examples are shown in (8).

(8) Sample Annotation Graphs for Russian and Chinese (Sproat, 2000)



These are graphs in the mathematical sense of the word "graph", in that the vertical lines represent vertices, and the horizontal lines represent arcs. In this notation, Sproat uses a colon to indicate the linguistic element which is responsible for licensing a particular grapheme, the idea being that both the grapheme and the element licensing it are attached to the same arc. This can be seen more clearly in (9).

(9) "Graph" form of second Annotation Graph in (8)

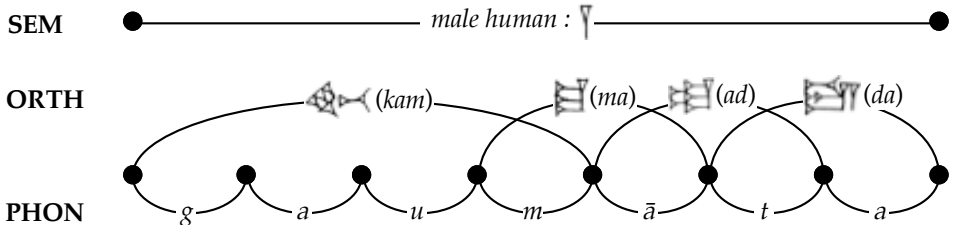


In most cases, the difference between Attribute Value Matrices and Annotation Graphs is simply one of notational convenience. However, Sproat's use of AVMs

to indicate licensing is not flexible enough to allow for cases (such as the example of *Gauma□ta* from §3.2) where a grapheme is licensed by multiple linguistic units, and where a single linguistic unit licenses portions of multiple graphemes. Any attempt to coindex graphemes and linguistic units would be fraught with difficulties.

However, the Annotation Graph notation does provide the necessary flexibility. Instead of writing the grapheme on the arc which licenses it, we add a whole separate set of arcs to indicate the orthography. Since we have seen that Elamite does not have a clean mapping between syllable structure and orthography, we will instead represent the phonological level as a sequence of phonemes. A grapheme is considered to be licensed by all the phonemes whose arcs are dominated by its arc, as shown in (10).

(10) Annotation Graph for *Gauma□ta* → *v.kam-ma-ad-da*



Annotation graphs of this sort are used as the internal representation for all the linguistic data being manipulated by the *Grotefend* software. Whenever the software has to evaluate a constraint or generate a rival form, it does so by performing operations on one or more annotation graphs.

4 Optimality Theory

As formulated by Prince & Smolensky (1993), Optimality Theory can be considered as a set of three interconnected modules: GEN, H-EVAL, and Lexicon Optimization. Together GEN and H-EVAL comprise the grammar proper. For any given input (In_k), GEN generates a set of output candidates. These output candidates or "rivals" are evaluated against a set of constraints by the H-EVAL module.

Lexicon Optimization is not part of the grammar, but it provides a mechanism by which language learners can use that grammar to determine underlying forms based on the overt forms which are presented to them. Prince & Smolensky's formulations for the three modules are set out in (11) and (12).

(11) GEN and H-EVAL (Prince and Smolensky, 1993)

$\text{GEN}(\text{In}_k) \rightarrow \{ \text{Out}_1, \text{Out}_2, \ldots \}$

$\text{H-EVAL}(\text{Out}_i, 1 \leq i \leq \infty) \rightarrow \text{Out}_{real}$

(12) Lexicon Optimization (Prince and Smolensky, 1993)

> Suppose that several different inputs In_1, In_2,…, In_n when parsed by a grammar G lead to corresponding outputs Out_1, Out_2, …, Out_n, all of which are realized as the same phonetic form Φ — these inputs are all *phonetically equivalent* with respect to G. Now one of these outputs must be the most harmonic, by virtue of incurring the least significant violation marks: suppose this optimal one is labelled Out_k. Then the learner should choose, as the underlying form for Φ, the input In_k.

As discussed by Tesar and Smolensky (2000), Optimality Theory can serve as a model of language learning. The language learner is presented with overt forms, and is able to determine both the constraint rankings and the underlying forms. At first, both the constraint rankings and underlying forms will be inaccurate, but as the learner is presented with more information the estimates of the underlying forms become more accurate, which in turn improves the constraint rankings, which further improves the estimates of the underlying forms, and so on. With this sort of iterative approach, the language learner is able to deduce the language's actual underlying forms and constraint rankings.

While this theory of learning is intended to describe the acquisition of spoken language, it can also be generalized to solve other problems. In particular, a modified version of this learning algorithm will form the basis of our strategy for reconstructing Elamite phonology.

4.1 Working with orthography

In the normal application of Optimality Theory, the input and the output are both the same type of linguistic entity. Thus, in the case of a typical phonology problem, both the input and the output would consist of a string of phonemes. However, in the case we are dealing with here, the relationship is between an input which is phonological and an output which is orthographic. The fact that we are comparing phonological apples to orthographic oranges leads to complications which will be discussed in §6.2.

In fact, we can think of there being four levels of representation lying between the input and the output. They are listed in (13). We can directly observe only level IV, but the hope is that we can reconstruct levels III and II. It seems likely that level I will remain largely inaccessible. From the perspective of our application of Optimality Theory to this problem, level IV contains the overt forms which are presented to our algorithms, while level II is what will be the output of Lexicon Optimization.

(13) Levels of representation

I	underlying phonology	The normal phonological input of Optimality Theory.
II	surface phonology	The normal phonological output of Optimality Theory.
III	underlying orthography	The writing system, as conceived by the Elamite scribe. So for instance, the scribe might think of the Akkadian *ṣi* grapheme as actually representing /tʃi/.
IV	surface orthography	The grapheme as it appears in the writing.

4.2 Applying Optimality Theory to Elamite

The overall strategy is to treat the reconstruction of Elamite phonology as a special type of learning problem. In a typical language acquisition situation as described by Tesar & Smolensky (2000), the learner is presented with surface phonology (level II) and is able to acquire the underlying phonology (level I). In this study, the "learner" is the *Grotefend* software, which is presented with surface orthography (level IV) and attempts to deduce level III and level II.

In order to "learn" Elamite phonology properly, all the modules of the Optimality Theory must be adapted for use with orthography.

First of all, the GEN module must be adapted to generate plausible overt forms (i.e. the rival orthographies of level IV). The general strategy for GEN is described in §5, with the specific details given in §5.3.

In its initial state, the constraint system will start out with equally-ranked constraints. The constraints have been designed to test the various hypotheses listed in §2.5.

The constraints are used by a ranking algorithm (§4.3 and §4.4) which scores the rival level IV orthographies from GEN against underlying phonological forms in order to determine the number of constraint violations. In order to start the process, those underlying forms have to be seeded with reasonable initial estimates. If the word is a loanword, the initial estimate is based on the Old Persian or Akkadian phonology. If there is no available loanword phonology, the initial estimate is merely a direct transcription of the grapheme values as if the word were being read in Akkadian.

Once the constraints have been ranked, the program can proceed to Lexical Optimization. This process takes the level IV forms and the newly-ranked constraints, and calculates an estimated phonology (level II) for each of the forms. At this point, the program can stop, or else it can proceed through another iteration of the ranking algorithm, using the new improved estimated phonologies as underlying forms.

4.3 Constraint Demotion

From the earliest days of Optimality Theory (Tesar, 1995) an algorithm called Constraint Demotion has been proposed to derive constraint rankings and structural descriptions given only the overt forms. The principle of Constraint Demotion is really quite simple. In any OT analysis one has a set of competing overt forms, referred to as candidates or rivals. Constraint Demotion consists of

comparing two candidates, cancelling off any matching markings, and reducing the ranking of any constraints which are causing problems. Or more formally:

(14) Constraint Demotion (Tesar, 1995)

For any constraint $\mathbb{C}$ assessing an uncancelled winner mark, if $\mathbb{C}$ is not dominated by a constraint assessing an uncancelled loser mark, demote $\mathbb{C}^*$ to immediately below the highest-ranked constraint assessing an uncancelled loser mark.

The idea is simply that if a constraint is too highly ranked and is picking the wrong winner, then that constraint should be demoted just far enough to make the real winner win.

Because Constraint Demotion is so straightforward, it has a number of provable properties. For instance, Tesar shows that for a system of N constraints, Constraint Demotion will require no more than N(N - 1) informative examples to determine the correct constraint rankings. It is not clear what N might be in a real language, but a reasonable intuition would place the number in the hundreds or in the thousands, rather than say in the millions.

Constraint Demotion does have a number of limitations. For one, all the constraints must be known in advance. In theory, this is not an issue, since CON, the set of all constraints, is part of Universal Grammar and will be the same across all languages; only the rankings change from language to language. In practice however, linguists have little agreement as to what constraints are universal.

A second limitation is that Constraint Demotion cannot cope with data for which there is optionality in the overt forms. That is, Constraint Demotion assumes that each underlying form will have exactly one overt form. If this is not the case, then Constraint Demotion can fall into an infinite loop, demoting one constraint and then another in a vain attempt to converge on the right winner.

4.4 Gradual Learning Algorithm

The difficulty faced by Constraint Demotion in the presence of optionality in overt forms is a serious one given the data from Elamite orthography. Not only are the orthographic forms subject to considerable variation, but also this variation is a key piece of information in attempting to reconstruct the phonology. Fortunately, Paul Boersma (1997) came up with a variation on Constraint Demotion called the Gradual Learning Algorithm (GLA), expanded in collaboration with Bruce Hayes (Boersma and Hayes, 2001).

The GLA is actually a fundamental reinterpretation of what it means for constraints to be ranked. Constraints each have a numeric ranking value associated with them. It is no longer the case that Constraint A consistently outranks Constraint B. Instead, whenever evaluating constraints, a random "noise" factor is added to each of the ranking values, and an instantaneous constraint ordering is determined based on these adjusted values. If the ranking values for two constraints are far apart, the chance of the noise altering the ordering is vanishingly small, and the results will be effectively the same as ordinary OT. However, if the ranking values for two constraints are close together, either constraint could potentially end up on top. Because of the random factor, the algorithm thus avoids the infinite looping problem faced by Constraint Demotion.

Furthermore, based on their empirical testing, Boersma & Hayes show that if you feed overt forms to the GLA in their real-world proportions, the GLA can derive a constraint ranking which will produce the observed forms in roughly the correct proportions. That being said, the GLA is not without its limitations. In particular, because it is probabilistic, the algorithm has to be run multiple times to produce results.

Prof. Hayes was kind enough to provide the Visual Basic source code which implements the GLA. The code was translated into C++ and adapted to work with the Elamite data which is the object of our study.

5 Implementation of GEN

Any linguist who has ever put together an OT tableau knows that one of the first thing to do is to come up with a plausible set of candidates which will all lose to the real form. As formulated by Prince & Smolensky (1993), that is the role of the GEN module.

However, if you look at their formulation in (11), it is worth noting that the subscript on Out_i goes from 1 to ∞. To quote Daniel Hall (2000), "a grammar that depends on the generation and evaluation of an infinite number of candidate forms is well-defined only at the most abstract level."

5.1 *Infinity Limited* (Hall, 2000)

There have been a number of attempts at taming GEN. In his discussion of prior work, Hall (2000) lists at least 6 earlier proposals: Ellison (1995), Hammond (1997), Karttunen (1998), Frank & Satta (1998), Eisner (1997), and Heiberg (1999).

Several of these papers are devoted to syllabification problems, and one would expect that GEN for a syllabification problem would produce a bounded set, simply by the nature of syllabification. After all, for any given string, there are only so many ways to syllabify it. However, if one allows for epenthesis, the set produced by GEN will be infinite, since OT on its own places no restriction on what sort of epenthetic material could be inserted. Common sense will limit the possibilities for epenthetic material to a manageable set, but strictly speaking, common sense is not part of Optimality Theory.

However, even common sense is sometimes inadequate. Hall points out that even a simple syllabification problem like the one which Prince & Smolensky (1993) provide to demonstrate OT (§8.3 of their paper) fails to find an answer that Hall's own algorithm does. The simple reason is that Prince & Smolensky were doing the work of GEN "by hand" and they left off a plausible candidate from their tableau.

Although Hall's paper presents a novel algorithm for solving the GEN problem, we will not discuss the algorithm in detail since it has a number of limitations that make it unsuitable for the purposes of the Elamite orthography problem. One restriction is that the algorithm depends on having no equally-ranked constraints, but the Elamite data may well require equally-ranked constraints to explain some of the observed spelling variation.

A more serious limitation is that Hall's algorithm for generating candidates requires that the ranking of constraints already be known, which creates a chicken and egg situation: we cannot determine the constraint rankings until we have some candidates, but we cannot generate the candidates until we know the constraint rankings.

This chicken and egg problem means that Hall's approach runs counter to the underlying claim that OT can be a learning model. The same limitation also holds for a number of the prior approaches mentioned by Hall, which are only able to generate a finite set of candidates if the constraint rankings are already known.

5.2 *Features in Optimality Theory* (Heiberg, 1999)

Of all the prior approaches listed by Hall, the most promising one was Heiberg (1999). Her approach to GEN is simple. From the input form, apply one of the four operations listed in (15).

(15) Operations available to GEN (Heiberg, 1999)

Insert Feature
Delete Feature
Insert Association
Delete Association

Like Hall, Heiberg's algorithm proceeds by choosing a starting point and then adding constraints to the system. As each constraint is added, new candidates are generated using what she calls "relevant" GEN operations. A GEN operation is considered to be relevant for the current constraint if the operation could affect

a candidate's harmony relative to that constraint. So for instance, if the constraint being added evaluates the [+back] feature, the only operations which are relevant are ones which affect [+back] or its associations.

As with Hall's algorithm, the candidates at each stage are not fully formed, and are slowly refined as the constraints are added to the system. However, while Hall's algorithm absolutely requires that the constraint ranking be known in order to operate, Heiberg's algorithm should be able to function even if the constraint rankings are not known. If the constraint rankings are known, the algorithm can operate more efficiently, by culling known losing candidates, but knowing the rankings is not essential.

5.3 Constraining GEN for Orthography

Heiberg's four operations are appropriate to autosegmental phonology, but they are not suitable for generating surface forms for this study. Nonetheless, the idea that candidates should be "relevant" to the constraint being evaluated is a useful one.

Remember that the purpose of GEN is to generate overt forms, and the overt forms in the Elamite problem are orthographic forms (i.e. level IV, as described in (13)). For any underlying form, the challenge is to generate orthographic strings which compete with the real orthography, but which are "wrong" with respect to one or more of the constraints.

The inventory of graphemes which are available to GEN is based on Steve (1992), which was intended to be the definitive work on the study of the Elamite syllabary throughout its history. That being said, Steve's work is not without its detractors, notably Stolper (p.c.). In particular, there are a number of disagreements between the readings provided by Hinz & Koch (1987) and those provided by Steve. Where necessary, the inventory given by Steve is expanded to include possibilities suggested by other authors.

The original software implementation of GEN for the Elamite problem used the notion of "mutation", where one of the four mutations shown in (16) would be

applied to an input form. Mutations were randomly applied until a desired number of candidate forms had been generated.

(16) Operations available to GEN for cuneiform
(examples for the word *bat-te-ra* 'herdsman')

Tweak	*bad-te-ra* *bat-ti-ra*,…	Replaces a grapheme with a phonetically "neighbouring" grapheme.
Deletion	*ba-te-ra* *bat-e-ra*	Replaces a *CVC* grapheme with a *CV* or *VC* grapheme; replaces a *CV* or *VC* grapheme with a *V* grapheme.
Merger	n/a	Replaces a *CV-VC* sequence with an appropriate *CVC* grapheme.
Split	*ba-at-te-ra* *bat-te-e-ra*	Replaces a *CVC* grapheme with a *CV-VC* sequence; replaces a *CV* grapheme with *CV-V* or a *VC* grapheme with *V-VC*.

While the results of mutation were interesting, it was felt that the randomly-generated candidates did not do a good job of actually testing the constraints. Some constraints received plenty of exercise, while other constraints were not tested at all.

The alternative approach, of generating all "plausible" candidates for a given form, was discarded as being computationally prohibitive. An algorithm was developed which took a source form and replaced its graphemes with phonetically neighbouring graphemes. That is, it applied all possible permutations of the Tweak operation from (16). Initial experiments indicated that a moderately long string of four graphemes would generate in the neighbourhood of 18000 rivals. It seemed unrealistic to evaluate tens of thousands of rivals for each of the 8000+ forms in the database, particularly since

the Gradual Learning Algorithm would require multiple iterations in order to converge on a constraint ranking.

Borrowing Heiberg's notion of "relevant" operations, the actual approach is to generate candidates which specifically exercise one of the constraints in the constraint system. So for instance, if the input form were *be-ul*, the broken vowel constraints (i.e. those constraints relating to the four hypotheses in group H1) would generate rival candidates such as *be-al*, *be-el*, and *be-il*. Each of the hypothesis groups listed in §2.5 refers to a particular orthographic context, and each of those contexts has a miniature version of GEN which generates appropriately test-worthy rivals. The output of each of these miniature GENs will be a set of rival outputs which resemble the correct one, but which will trigger a different set of constraint violations.

The details of the various mini-GENs are as follows:

H1) Broken vowels

> **Rule:** Whenever a *CV-VC* sequence is found in the orthography, rivals are generated by permuting the second vowel.
> **Example:** *da-iš* → { *da-aš, da-áš, da-eš, da-iš, da-uš* }

H2) Voicing of stops

> **Rule:** Whenever a grapheme is encountered with the value of a voiced stop, rivals are generated by permuting all the graphemes containing the corresponding voiceless values, and *vice versa*.
> **Example:** *da-iš* → { *da-iš, ta-iš, tà-iš* }
> *te-em-ti* → { *dé-em-di, dé-em-ti, te-em-di, te-em-ti* }

H3) Geminate consonants

> **Rule:** Whenever a geminate consonant is found in the orthography, generate a rival with the non-geminate equivalent.
> **Example:** *hu-ut-ta* → { *hu-ta* }

H4) Nasal vowels

Rule: Whenever a non-intervocalic nasal consonant is found in the orthography, generate rivals which have the equivalent orthography without the nasal segment. This can be done both by converting the nasal to an oral stop with the same place of articulation, or by eliminating the nasal segment altogether.
Example: *te-em-ti* → { *te-eb-ti*, *te-ti* }

H5) Word-final vowels

Rule: Whenever a word-final vowel is found in the orthography, generate rivals which have the equivalent orthography, but with all the permutations of final vowels. Also, where possible, generate a rival with the equivalent orthography, but without a final vowel.
Example: *hu-ut-ti-be* → { *hu-ut-ti-ba, hu-ut-ti-be, hu-ut-ti-bi, hu-ut-ti-bu, hu-ut-ti-ib* }

H6) Sibilants

Rule: Whenever a sibilant is found in the orthography, generate rivals which have the equivalent orthography, but with all permutations of sibilant signs.
Example: *su-un-ki* → { *su-un-ki, šu-un-ki, šú-un-ki, zu-un-ki* }

H7) The phonemic inventory includes an /h/.

Rule: Whenever an /h/ is found in the orthography, generate rivals which have the equivalent orthography but without the /h/.
Example: *hu-ut-ta* → { *u-ut-ta, ú-ut-ta, ù-ut-ta* }

H8) The phonemic inventory includes /f/ and /v/.

Rule: None.
Comment: Since the grapheme inventory lacks signs with /f/ and /v/ values, it is difficult to construct a useful set of rivals to test this particular context.

H9) The *ya* grapheme indicates a /j/

Rule: Whenever a /j/ is found in the orthography (in the form of a *ya* grapheme), generate a rival which has the equivalent orthography without the /j/.
Example: *hi-ya-an* → { *hi-a-an, hi-an* }

H10) The *ú* grapheme indicates a /w/

Rule: Whenever the *ú* grapheme is encountered, generate a rival which omits it.
Example: *hh.da-a-hi-ú-uk-qa* → { *hh.da-a-hi-uk-qa* }
Comment: This rule is based on McAlpin (1982) who makes a specific claim about the use of the *ú* grapheme to indicate a /w/.

H11) /e/

Rule: Whenever an /i/ is found in the orthography, replace it with the equivalent grapheme which has an /e/ value, and *vice versa*.
Example: *te-em-ti* → { *ti-im-ti, ti-im-te, te-em-te* }
Comment: Note that the candidates listed here do not include obvious choices such as *te-im-te* or *te-im-ti* because the  grapheme is used for both *im* and *em*. Borrowing from Akkadian practice, there are many Elamite graphemes which do not distinguish between /e/ and /i/. Here, as elsewhere, the choice of rival candidates is constrained by the inventory of available graphemes.

6 Implementation of H-EVAL

The H-EVAL module is responsible for the actual evaluations of the various candidates. As such, it has to be able to take any output candidate produced by GEN, and count the violation marks for each of the constraints in the system.

6.1 General implementation of constraints

The core of the implementation of each constraint is a C++ function called `violations()`. This function has two inputs (an underlying form and a surface form), and produces as an output the number of violations incurred by the comparing the two inputs. For full generality, both inputs are annotation graphs (§3.3), the idea being that constraint evaluation could conceivably involve the semantic and morphological portions of the annotation graph. As implemented in the *Grotefend* software, the actual comparison typically involves only the phonology portion of the underlying form's graph and the orthography portion of the surface form's graph.

The constraints used for this project were designed to test the various hypotheses presented in §2.5. Since there is no prior art in the area of constraints involving orthography and phonology, they were developed in the most straightforward way possible. That is, each hypothesis was considered, and an appropriate `violations()` function was written for which would score a violation whenever the hypothesis turned out to be incorrect. These scoring rules are summarized in §6.3.

It should be pointed out that many of the constraints are rather simple-minded, and can potentially produce misleading results. It is hoped that with further research into the problem of applying OT to orthography, it should be possible to develop more sophisticated constraints which better capture the complexities of the $M_{ORL \rightarrow \Gamma}$ mapping.

6.2 Implementation of alignment

In order to invoke the `violations()` function, the two inputs must be properly aligned. If an annotation graph is "aligned", it means that we have a mapping between phonology and orthography, where every grapheme is licensed by some portion of the phonology, and every phoneme is represented in the orthography. Without such a mapping, it is impossible to compare the phonology against the orthography in order to score constraint violations.

The problem of alignment turns out to be quite challenging, and a number of approaches were tried and then discarded. The most sophisticated of these involved calculating the phonetic "distance" between each grapheme and the phonemes which licensed it, and trying to minimize that distance.

In the end, the best solution was simply to follow Voltaire's dictum that the consonants count for very little, and the vowels for nothing at all. The most effective approach was simply to line up the consonants and let the vowels fall in where they may. For an example, consider the Elamite spelling of the Zoroastrian divinity Ahuramazda□h, which appears in a number of orthographies, including *d.u-ri-um-maš-da*. Using the Old Persian phonology as the best available initial estimate for the Elamite phonology (level II), the stages to be followed in licensing the orthography (level IV) are shown in (17).

(17) Licensing process for Ahuramazda□h → *d.u-ri-um-maš-da*

d is a determinative, and is licensed by the semantic tier of the annotation graph, so it does not need to be anchored to the phonology.

u is anchored at the left edge of the phonology.

ri starts at phoneme /r/, but has no clear right edge. The anchoring of /r/ sets a right boundary on the *u*, which must therefore be licensed by the initial /ahu/ of /ahuramazda:h/.

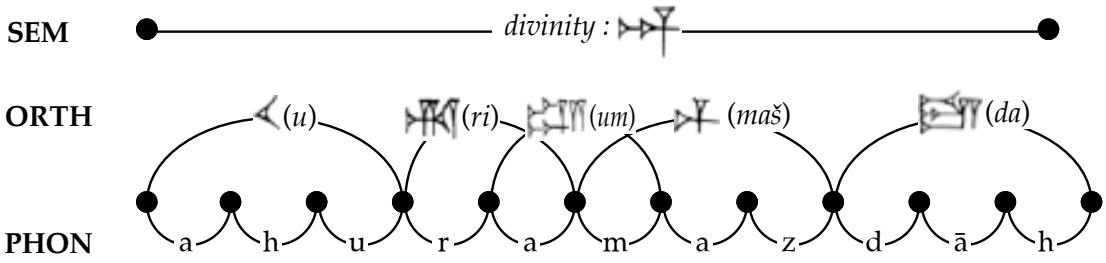
um right edge at phoneme /m/; since the *um* has no clear left edge, the second /a/ of /ahuramazda:h/ is left floating between the *ri* and the *um*. Since there is no clear choice between the two locations, the /a/ will be shared by *ri* and *um*.

maš starts at phoneme /m/ and ends at phoneme /z/, which is an adequate match for *š*. The /m/ will be shared by *um* and *maš*.

da starts at phoneme /d/; since *da* is the last grapheme, it must be licensed by the remainder of the phonology.

At the end of this process, the program has been able to calculate the annotation graph shown in (18). Only after the annotation graph has been calculated and the licensing determined, is it possible to apply the constraints detailed in §6.3. So for instance, this particular graph could now be evaluated for the broken-vowel constraints (since the orthographic tier contains the CV_1-V_2C sequence *ri-um*). The software would find violations for H1a (level II consists of two separate vowels), H1b (level II contains a vowel-glide sequence), and H1d (level II has phonological vowel with a value of V_1), but no violation for H1c (level II uses a vowel different from V_1 and V_2).

(18) Annotation graph calculated for Ahuramazda□h → *d.u-ri-um-maš-da*



In order to decide whether two consonants are eligible to be matched up (as can be seen in (17) for the matching of *š* and /z/ in the *maš* phoneme), there must be a way to calculate the "distance" between two phonemes. The approach taken was to assign a weight to each phonological feature, and to say that the distance between two phonemes is the sum of the weights for all features that differ between the two phonemes. The full listing of feature weights is shown in (19).

The weighting values were determined through experimentation, selecting the weightings which did the best job of aligning the orthography for the Old Persian loanwords given by Hinz (1987).

(19) Weights of phonological features for computing "distance"

Feature	Weight
delayed release voice labio-dental anterior distributed strident	1
approximant continuant nasal lateral round low pharyngeal	2
syllabic consonantal constricted glottis spread glottis high front back	4
sonorant place of articulation	8

The situation is further complicated when the algorithm has to align phonologies which contain /h/ and /w/ phonemes which are not well-represented in Elamite cuneiform. However, the general strategy of aligning consonants seems to be an effective one. In the working dataset of Achæmenid Elamite words, there were 3045 which used Old Persian or Akkadian data to provide an initial estimate of the underlying phonology. The algorithm successfully aligned 2902 of those words, for a success rate of over 95%. Since words whose orthography cannot be aligned must be excluded from further processing, it is important to get this success rate as high as possible.

6.3 Implementation of constraints for testing hypotheses

Assuming that the orthography can be successfully aligned with the underlying phonology, it is possible to score the forms for violations against all the constraints in the system.

In terms of the levels described in (13), the constraints are performing comparisons between underlying forms from level II and overt forms from level IV. In terms of the tiers shown in annotation graphs like (10) and (18), the comparisons are being made between the underlying forms in the PHON tier and overt forms in the ORTH tier. In theory, the SEM tier could also participate in the evaluation of constraints, but that is not necessary given the particular set of hypotheses we are trying to evaluate. The specific rules for calculating constraint violations for each of the hypotheses in §2.5 are as follows:

H1a) V_1 and V_2 are articulated as separate vowels.

> **Rule:** Score a violation whenever the orthography contains a CV_1-V_2C sequence if:
>
> 1) the underlying phonology for /V_1/ equals /V_2/ or
> 2) either /V_1/ or /V_2/ is not a vowel.
>
> **Example violations:** /daʃ/ → *da-iš*, /dajʃ/ → *da-iš*
>
> **Example non-violations:** /daiʃ/ → *da-iš*
>
> **Comment:** Although in a real language, it might be difficult to make a distinction between /dajʃ/ and /daiʃ/, the software has no such difficulty; as far as the program is concerned, the two are clearly distinct, since the syllabic feature of /j/ has a value of –1 while the syllabic feature of /i/ has a value of +1. This is one example of how the software's simplified model of phonology can lead it to make a clear-cut distinction which is not justified in the real world.

H1b) V_1 and V_2 represent a diphthong.

> **Rule:** Score a violation whenever the orthography contains a CV_1-V_2C sequence if:

1) the underlying phonology for /V_1/ equals /V_2/ or
2) /V_2/ is not a semivowel.

Example violations: /daʃ/ → *da-iš*, /daiʃ/ → *da-iš*

Example non-violations: /dajʃ/ → *da-iš*

H1c) V_1 and V_2 are articulated as an intermediate vowel.

Rule: Score a violation whenever the orthography contains a CV_1-V_2C sequence if:

1) the underlying phonology for /V_1/ is not equal to /V_2/ or
2) either /V_1/ or /V_2/ is not a vowel or
3) /V_1/ is equal to the vowel value from the CV_1 grapheme.

Example violations: /daiʃ/ → *da-iš*, /dajʃ/ → *da-iš*, /daʃ/ → *da-iš*

Example non-violations: /dɛʃ/ → *da-iš*

H1d) V_2 is ignored.

Rule: Score a violation whenever the orthography contains a CV_1-V_2C sequence if:

1) the underlying phonology for /V_1/ is not equal to /V_2/ or
2) either /V_1/ or /V_2/ is not a vowel or
3) /V_1/ is not equal to the vowel value of the CV_1 grapheme.

Example violations: /daiʃ/ → *da-iš*, /dajʃ/ → *da-iš*, /dɛʃ/ → *da-iš*

Example non-violations: /daʃ/ → *da-iš*

H2a) The choice of graphemes with voiced or unvoiced values reflects a voicing distinction.

Rule: Score a violation whenever:

1) a voiced stop in the orthography corresponds to a voiceless stop in the phonology or
2) a voiceless stop in the orthography corresponds to a voiced stop in the phonology.

Example violations: /giri/ → *ki-ri*, /pariʃda/ → *ba-ri-iš-da*,

Example non-violations: /giri/ → *gi-ri*

H2b) The choice of graphemes with voiced or unvoiced values reflects a distinction other than voicing (e.g. tense/lax).

Rule: Score a violation whenever:

1) a voiced stop in the orthography corresponds to a "tense" stop in the phonology or
2) a voiceless stop in the orthography corresponds to a "lax" stop in the phonology.

Example violations: /k'iri/ → *gi-ri*, /kiri/ → *ki-ri*

Example non-violations: /k'iri/ → *ki-ri*

H2c) The choice of graphemes containing voiced and voiceless values does not reflect a distinction in the phonology.

Rule: None. This hypothesis is simply the denial of hypotheses H2a and H2b.

H2d) Voicelessness is indicated by combining the [-voice] feature of one grapheme with the place of articulation of another.

Rule: Score a violation whenever the orthography contains a VC_1-C_2V sequence where one of C_1 and C_2 is voiceless, and C_1-C_2 does not correspond to a voiceless stop in the phonology.

Example violations: /kabar/ → *kak-bar*

Example non-violations: /upaka:ma/ → *uk-ba-qa-ma*

Comment: Hinz & Koch's claim is not that all voiceless stops are indicated with this sort of mechanism, only that whenever this mechanism does occur, it implies a voiceless stop in the phonology.

H3a) Geminate spellings represent geminate pronunciations.

Rule: Score a violation if the orthography contains a geminate consonant which does not correspond to a geminate in the phonology.
Example violations: /ata/ → *at-ta*
Example non-violations: /atta/ → *at-ta*, /atta/ → *a-ta*
Comment: The specific claim made by Grillot-Susini (1988) was only that a geminate orthography represents a geminate phonology; a non-geminate orthography could still conceal a geminate phonology.

H3b) Geminate spellings are used to indicate a voicing distinction.

Rule: Score a violation if:

1) the orthography contains an intervocalic geminate stop which does not correspond to a voiceless stop in the phonology or
1) the orthography contains an intervocalic non-geminate stop which does not correspond to a voiced stop in the phonology or
3) the phonology contains an intervocalic voiceless stop which does not correspond to a geminate in the orthography or
4) the phonology contains an intervocalic voiced stop which does not correspond to a non-geminate in the orthography or

Example violations: /duba:la/ → *du-ib-ba-la*,
/garmapada/ *d.kar-ma-ba-taš*
Example non-violations: /gauma:ta/ → *kam-ma-ad-da*,
/babili/ → *ba-pi-li*
Comment: Reiner (1969) restricted her claim about gemination representing voicelessness to intervocalic stops. Word-initial stops and intervocalic non-stops were not covered by her rule.

H3c) Certain geminate spellings are used to indicate a distinction other than voicing, such as retroflex/alveolar.

Rule: Score a violation if:

1) the orthography contains an *l-l* or *r-r* sequence which does not correspond to a "retroflex" in the phonology or
2) the phonology contains a /ɭ/ or /ɻ/ which does not correspond to an *l-l* or *r-r* sequence in the orthography.

Example violations: /talu/ → *ta-al-lu,*
Example non-violations: /taɭu/ → *ta-al-lu*

H4a) The observed alternations in the writing of nasals indicate the presence of nasal vowels.

Rule: Score a violation if there is a *VN-CV* sequence in the orthography and the nasal portion of the *VN* grapheme corresponds to a consonant in the phonology.
Example violations: /temti/ → *te-em-ti*
Example non-violations: /tẽti/ → *te-em-ti*

H4b) The observed alternations in the writing of nasals can be explained by underlying nasal consonants.

Rule: Score a violation if there is a *VN-CV* sequence in the orthography and the nasal portion of the *VN* grapheme does not correspond to a consonant in the phonology.
Example violations: /tẽti/ → *te-em-ti*
Example non-violations: /temti/ → *te-em-ti*

H5a) The alternations in the writing of word-final vowels indicate an attempt to render a word-final consonant cluster which could not otherwise be written.

Rule: Score a violation whenever the orthography ends with a *VC-CV*$_2$ sequence if the phonology ends with a vowel.
Example violations: /temti/ → *te-em-ti*
Example non-violations: /temt/ → *te-em-ti*

H5b) The alternations in the writing of word-final vowels indicate the presence of a /ə/ or other underspecified vowel.

> **Rule:** Score a violation whenever the orthography ends with a *CV* grapheme if:
>
> 1) the phonology ends with a consonant or
> 2) the phonology ends with a vowel which is identical to the value of *V*.
>
> **Example violations:** /temt/ → *te-em-ti*, /temti/ → *te-em-ti*
> **Example non-violations:** /temtə/ → *te-em-ti*

H5c) The alternations in the writing of word-final vowels indicate an actual alternation in the vowel.

> **Rule:** Score a violation whenever the orthography ends with a *CV* grapheme if:
>
> 1) the phonology ends with a consonant or
> 2) the phonology ends with a vowel which differs from the value of *V*.
>
> **Example violations:** /temt/ → *te-em-ti*, /temtə/ → *te-em-ti*
> **Example non-violations:** /temti/ → *te-em-ti*

H6a) The sibilant inventory includes a /tʃ/ which is written using the Akkadian *ṣV* and *Vṣ* graphemes

> **Rule:** Score a violation if:
>
> 1) the orthography contains an *ṣV* or *Vṣ* grapheme whose consonant corresponds to something other than a /tʃ/ in the phonology or
> 2) there is a /tʃ/ in the phonology which corresponds to something other than an *ṣV* or *Vṣ* in the orthography
>
> **Example violations:** /zalmu/ → *ṣa-al-mu*, /□p□a□ntʃu□k□aʃ/ → *pan-su-kaš*
> **Example non-violations:** /bagaitʃa/ → *hh.ba-gi-iṣ-ṣa*

H6b) The sibilant inventory includes a /z/ which is written using the Akkadian *ṣV* and *Vṣ* graphemes

Rule: Score a violation if:

1) the orthography contains an *ṣV* or *Vṣ* grapheme whose consonant corresponds to something other than a /z/ in the phonology or
2) there is a /z/ in the phonology which corresponds to something other than an *ṣV* or *Vṣ* in the orthography

Example violations: /anʃan/→*an-ṣa-an*, /bagaba:zu/→*hh.ba-qa-ba-su*

Example non-violations: /zalmu/→*ṣa-al-mu*

Comment: The combination of constraints for H6a and H6b precludes the possibility that there are separate phonemes for /z/ and /tʃ/, both of which would be rendered using *ṣV* and *Vṣ* graphemes. However, we will omit this possibility since none of the previous scholars have proposed a phonemic inventory which includes both /z/ and /tʃ/.

H6c) The sibilant inventory includes a /ts/ affricate which is written using the *sV*, *šV*, *tV*, *Vs*, *Vš*, or *Vt* graphemes.

Rule: Score a violation if the phonology contains a /ts/affricate which does not correspond to a *s*, *š*, or *t* in the orthography.

Example non-violations: /tsemti/→*te-em-ti*

Comment: This rule is less penalizing than the ones for H6a and H6b, for the simple reason that we do not want to preclude the existence of /s/, /ʃ/, and /t/ phonemes which would share the use of same graphemes as /ts/.

H7a) The phonemic inventory includes an /h/, which is written using the *hV* and *Vh* graphemes.

Rule: Score a violation if:

1) the orthography contains an *hV* or *Vh* grapheme which does not correspond to an /h/ in the phonology or

2) there is an /h/ in the phonology which corresponds to something other than an *hV* or *Vh* in the orthography

Example violations: /huttip/ → *ú-ut-ti-ip,* /api/ → *a-ah-pi*

Example non-violations: /huttip/ → *hu-ut-ti-ip*

H7b) The *hV* and *Vh* graphemes are merely orthographic variants of the corresponding *V* graphemes.

Rule: Score a violation if the orthography contains an *hV* or *Vh* grapheme which corresponds to an /h/ in the phonology

Example violations: /huttip/ → *hu-ut-ti-ip*

Example non-violations: /api/ → *a-ah-pi*

H8a) The *pír* grapheme is being used to indicate a /fr/ or /vr/ sequence (Khachikjan, 1998).

Rule: Score a violation if:

1) the orthography contains a *pír* grapheme which does not correspond to an /fr/ or /vr/ sequence in the phonology or
2) the phonology contains an /fr/ or /vr/ sequence which does not correspond to a *pír* grapheme.

Example violations: /hambrta/→*hh.am-pír-da*

Example non-violations: /fraʃam/→*hh.pír-ra-šá-um*

H8b) The *pír* grapheme is being used to indicate a /pr/ or /pir/ sequence.

Rule: Score a violation if the orthography contains a *pír* grapheme which does correspond to an /fr/ sequence in the phonology.

Example violations: /fraʃam/→*hh.pír-ra-šá-um*

Example non-violations: /hambrta/→*hh.am-pír-da*

H9a) The *ya* grapheme is being used to write a /j/

Rule: Score a violation if the orthography contains a *ya* grapheme which does not correspond to a /j/ in the phonology.
Example violations: /iaunap/→*ya-u-na-ap*
Example non-violations: /jaunap/→*ya-u-na-ap*
Comment: As mentioned above in the discussion for H1a, the distinction between /j/ and /i/is an easy one for software to make, but somewhat more difficult to establish in an actual language. The hope here was that there would be Old Persian loanwords which made a clear distinction between /i/ and /j/, and that the use of *ya* in those words might provide insight into the existence of a /j/. It should be mentioned though that the distinction in Old Persian is not as secure as it might be. The loanword most commonly cited as evidence of a /j/ in Elamite (Paper, 1955) is the Old Persian word *yauna* 'Greek', but it is worth noting that the same stem manifests itself in English as Ionian, and not as Yonian.

H9b) The *ya* grapheme is being used to write a non-syllabic allophone of /i/.

Rule: Score a violation if the orthography contains a *ya* grapheme which does not correspond to an /i/ in the phonology.
Example violations: /jaunap/→*ya-u-na-ap*
Example non-violations: /iaunap/→*ya-u-na-ap*

H10a) The grapheme *ú* is being used to indicate a /w/ (McAlpin, 1982).

Rule: Score a violation if:
1) the orthography contains a *ú* which does not correspond to a /w/ in the phonology or
2) the phonology contains a /w/ which does not correspond to a *ú* in the orthography.

Example violations: /dahiwukka/→*hh.da-a-hi-uk-qa*
/dahju:kka/→*hh.da-a-hi-ú-uk-qa*

Example non-violations: /dahiwukka/→*hh.da-a-hi-ú-uk-qa*

Comment: The same issues with the /i/ vs. /j/ distinction can be raised with the /u/ vs. /w/ distinction. However, McAlpin's claim is specifically related to the use of the *ú* grapheme.

H10b) The grapheme *ú* is being used to indicate a /u/.

Rule: Score a violation if the orthography contains a *ú* which does not correspond to a /u/ in the phonology.

Example violations: /dahiwukka/→*hh.da-a-hi-ú-uk-qa*

Example non-violations: /dahju:kka/→*hh.da-a-hi-ú-uk-qa*

H11a) There is an /e/ vowel, distinct from /i/.

Rule: Score a violation if:

1) the orthography contains an *e, eC, or Ce* grapheme whose vowel does not correspond to an /e/ in the phonology or
2) the phonology contains an /e/ which does not correspond to an *e, eC, or Ce* grapheme.

Example violations: /hutip/→*hu-te-ip*, /temti/→*ti-im-ti*

Example non-violations: /alumelu/→*a-lu-me-lu*

H11b) There is an /e/ vowel, distinct from /i/, but only in the word-initial syllable (McAlpin, 1982).

Rule: Score a violation if:

1) the initial syllable of the orthography contains an *e, eC, or Ce* grapheme whose vowel does not correspond to an /e/ in the phonology or
2) the initial syllable phonology contains an /e/ which does not correspond to an *e, eC, or Ce* grapheme.

Example violations: /temti/→*ti-im-ti*

Example non-violations: /hutip/→*hu-te-ip*

H11c) The *e, eC,* and *Ce* graphemes are merely orthographic variants of the corresponding *i, iC,* and *Ci* graphemes.

> **Rule:** Score a violation if the orthography contains an *e, eC,* or *Ce* grapheme which corresponds to an /e/ in the phonology
> **Example violations:** /temti/→*te-em-ti*
> **Example non-violations:** /timti/→*te-em-ti*

7 Implementation of Lexicon Optimization

Given the set of constraints provided in §6.3 and rankings determined by the Gradual Learning Algorithm (§4.4), it is now possible to move on to the final stage of the "learning" process: Lexicon Optimization. This is the module which is responsible for actually determining the phonological forms (i.e. level II in our schema).

As stated in (12) above, the Lexicon Optimization module is intended to choose between several input forms which are realized as the same phonetic form, Φ. Since the surface forms in the Elamite problem are orthographic and not phonetic, (12) must be restated as follows:

(20) Lexicon Optimization for Orthography

> Suppose that several different inputs In_1, In_2,…, In_n when parsed by a grammar G lead to corresponding outputs Out_1, Out_2, …, Out_n, all of which are realized as the same orthographic form Γ — these inputs are all *orthographically equivalent* with respect to G. Now one of these outputs must be the most harmonic, by virtue of incurring the least significant violation marks: suppose this optimal one is labelled Out_k. Then the program should choose, as the estimated underlying form for Γ, the input In_k.

Compared to the rest of Optimality Theory, there has been surprisingly little literature devoted to Lexicon Optimization. Prince & Smolensky (1993) introduced the concept along with the rest of Optimality Theory. Tesar &

Smolensky (2000) devote a brief section to Lexicon Optimization as part of their discussion of Optimality Theory as a theory of learnability. The survey of OT literature done by McCarthy (2002) confirms that Lexicon Optimization has received very little attention compared to other parts of OT.

Those papers which have discussed Lexicon Optimization, such as Itô et al (1995), have dealt with Lexicon Optimization on a very philosophical level, concerning themselves such as the issue of what exactly is being optimized (i.e. does the output Lexicon Optimization consist of full or unspecified forms?). As it turns out, the version of Lexicon Optimization implemented in *Grotefend* does result in underspecified forms, but this is not a philosophical choice, but simply a consequence of the fact that the software does not have sufficient information to derive fully-specified forms. In any case, such discussion of how Lexicon Optimization might actually be implemented are restricted to toy algorithms such as Itô's "tableau des tableaux". Such algorithms may have explanatory value, but they do not provide an adequate starting point for a software implementation.

Since there appears to be nothing in the way of usable concrete implementations of Lexicon Optimization, a novel approach had to be devised. The basic strategy was based on the observation that Lexicon Optimization is a sort of mirror image of H-EVAL. As set out in (12) above, for each input form, H-EVAL takes a range of possible outputs and selects the most harmonic one. Lexicon Optimization, does the opposite: for each output form, Lexicon Optimization takes a range of possible inputs and selects the most harmonic one.

In the case of H-EVAL, there exists a separate GEN module whose task is to generate the possible output candidates. Clearly, Lexicon Optimization has need of an equivalent module, but one which would generate a range of possible input forms. Since the GEN algorithm described in §5.3 uses a constraint-driven technique for generating output candidates, it seems appropriate to also use a constraint-driven technique for generating input candidates. Accordingly, this anti-GEN is implemented as a set of miniature anti-GENs, each of which is

responsible for generating "relevant" input candidates for the one of the eleven hypothesis groupings.

H1) Broken vowels

> **Rule:** Whenever a CV_1-V_2C sequence is found in the orthography, create input candidates which are appropriate to each of the four hypotheses (H1a, H1b, H1c, and H1d).
>
> **Example:** *da-iš* → { /daiʃ/, /dajʃ/, /dɛʃ/, /daʃ/ }

H2) Voicing

> **Rule:** Whenever a grapheme is encountered with a stop value, create input candidates with voiced and voiceless stops.
> **Example:** *da-iš* → { /daiʃ/, /taiʃ/ }
> *te-em-ti* → { /temti/, /temdi/, /demti/, /demdi/ }

H3) Geminate consonants

> **Rule:** Whenever a geminate consonant is found in the orthography, create input candidates with the geminate phonology and the equivalent non-geminate phonology. If the geminate orthography is an *r-r* or *l-l* sequence, also create an input candidate with a "retroflex" phonology.
> **Example:** *ta-al-lu* → { /tallu/, /talu/, /taɭu/}
> **Comment:** McAlpin (1982) hedges on whether the phonology represented by these geminates actually represents retroflexion, but he then proceeds to discuss Proto-Elamo-Dravidian cognates as if this orthography actually did represent a retroflex articulation. Khajickjan (1998) is not so ambivalent, and just describes it as retroflexion.

H4) Nasal vowels

Rule: Whenever a non-intervocalic nasal consonant is found in the orthography, create one input candidate which has the nasal consonant in the phonology and another input candidate which lacks the consonant but which has a nasalized vowel.
Example: *te-em-ti* → { /temti/, /tẽti/ }

H5) Word-final vowels

Rule: Whenever a word-final vowel is found in the orthography, create one input candidate which honours the phonology of the final vowel, one input candidate which has a final /ə/, and a third input candidate which has no final vowel.
Example: *hu-ut-ti-be* → { /hutibe/, /hutibə/, /hutib/ }

H6) Sibilants

Rule: Whenever a sibilant is found in the orthography, create input candidates which have the various sibilant fricatives and affricates.
Example: *su-un-ki* → { /sunki/, /ʃunki/, /zunki/, /tʃunki/, /tsunki/ }

H7) /h/

Rule: Whenever a grapheme whose value contains an /h/ is found in the orthography, create input candidates with and without the /h/.
Example: *hu-ut-ta* → { /hutta/, /utta/ }

H8) /f/ or /v/

Rule: Whenever a *pír* grapheme is found in the orthography, create input candidates with /fr/, /pr/, and /br/ sequences.
Example: *hh.pír-ra-šá-um* → { /fraʃaum/, /praʃaum/, /braʃaum/ }

H9) /j/

Rule: Whenever a *ya* grapheme is found in the orthography, create one input candidate with a /j/, one with a /i/, and one with neither.
Example: *ya-u-na-ap* → { /jaunap/, /iaunap/, /aunap/ }

H10) /w/

Rule: Whenever a *ú* grapheme is found in the orthography, create one input candidate with a /w/, one with a /u/ and one with neither.
Example: *ú-el* → { /wel/, /uel/, /el/ }

H11) /e/

Rule: Whenever an /e/ value is found in the orthography, create input candidates with /e/ and /i/ in the phonology.
Example: *te-em-ti* → { /temti/, /timti/ }

8 Results and Discussion

The *Grotefend* software applied 40000 iterations of the Gradual Learning Algorithm (Boersma and Hayes, 2001) to the Achæmenid Elamite forms found in the *Elamisches Wörterbuch*. The GLA portion of the program's execution took approximately four hours on a Macintosh with a 1GHz G4 processor. The final constraint rankings are shown in (21).

The number 40000 was chosen for the number of iterations based on a trade-off between execution time and the need to ensure that all the observed forms would have a chance to contribute to the final result. The working data-set consisted of 3748 unique words with a total of 5747 observed orthographies. With 40000 iterations, each observed orthography will be presented to the Gradual Learning Algorithm approximately 7 times.

In the GLA implementation implemented within the *Grotefend* software, every constraints starts with a ranking value of 100.00. With each iteration of the algorithm, one of the observed forms is selected as an exemplar, and rivals

(produced by GEN) are compared against the observed exemplar form. Whenever a rival beats the exemplar form, the constraint ranking values must be adjusted: all constraints which picked the wrong winner are penalized (adjusted downwards), and all constraints which picked the right winner are rewarded (adjusted upwards). The size of this adjustment is determined by a variable called "plasticity", which starts at 2.00, and is reduced gradually to 0.002 as the program proceeds through its iterations.

As can be seen in (21), the combination of constraints, GEN, and anti-GEN functions used by *Grotefend* tends to penalize constraints much more often than it rewards them. The absolute ranking values are not significant; what matters is their relative values.

(21) Constraint Rankings Produced by the Gradual Learning Algorithm

Hypothesis	Constraint	Ranking Value
H10b	<ú>=/u/	-53.03
H2d	Hinz	-77.03
H4b	NasalConsonants	-136.93
H9a	<ya>=/ja/	-149.02
H9b	<ya>=/ia/	-151.62
H11b	<e>=/é/	-185.31
H8b	<pír>=/pr/	-237.21
H3b	<Geminate>=/Voiceless/	-283.70
H8a	<pír>=/fr/	-346.71
H7a	<h>=/h/	-347.48
H7b	<h>≠/h/	-348.56
H5c	FinalVowel	-657.11
H10a	<ú>=/w/	-658.77
H11a	<e>=/e/	-659.50
H2a	<Voicing>=/Voicing/	-673.38
H6a	<ṣ>=/ʧ/	-937.31
H6b	<ṣ>=/z/	-938.14
H11c	<e>=/i/	-997.34
H4b	NasalVowels	-1434.77
H1d	<V1V2>=/V1/	-1435.65
H6c	Ǝ/ʦ/	-1436.05
H2b	<Voicing>=/Tense/	-1436.83
H3c	<Geminate>=/Retroflex/	-1629.74
H1a	<V1V2>=/V1V2/	-1629.98
H1b	<V1V2>=/Diphthong/	-2822.40
H1c	<V1V2>=/V1.5/	-2878.12
H5a	FinalCluster	-3045.34
H3a	<Geminate>=/Geminate/	-3189.11
H5b	FinalSchwa	-4043.82

Following the completion of the GLA portion, the Lexicon Optimization process was applied to the forms, which took an additional five hours of processing time. The output of this stage appears rather cryptic, since many of the estimates contain underspecified phonemes, as seen in (22). Each value in square brackets represents a bundle of features which does not have a corresponding IPA character, and in order to interpret the output one needs to know that the order of features is { syllabic, consonantal, approximant, sonorant, continuant, nasal, lateral, delayed release, voice, constricted glottis, spread glottis, LABIAL, round, labio-dental, CORONAL, anterior, distributed, strident, DORSAL, high, low,

front, back, tense, PHARYNGEAL }. Thus, it is possible to figure out that the third phoneme in the estimate for *hh.šu-si-qa* is some sort of [-anterior] coronal sibilant. Clearly, there needs to be a better way of presenting this sort of data in human-readable form, but it is not obvious how to do so.

(22) Sample output of Lexicon Optimization

Optimized θuθ īk□a { hh.šu-si-qa } to
ʃu[-+--+------- + +- --]i[-+---------- - ++- -]a
Optimized ṣalmu { za-ul-man } to
zaulman
Optimized ṭab-ili { hh.da-pi-li? } to
[-+------ --- ++--- +-]a[-+------ --+--- - --][+-+++---+--- - + -+---]li

8.1 General discussion of constraints

The specific results for each of the constraints will be discussed in §8.3, but there are a number of issues which are relevant to all the constraints in the system.

It was clear that there was at least one significant limitation which applies across all constraints. All the constraints are scored based on the comparison of phonology versus orthography in constraint-specific contexts. So for instance, the broken vowel constraints are only evaluated where there is a CV_1-V_2C sequence in the orthography, while the nasal vowel constraints are only evaluated where there is a *VN-CV* sequence in the orthography. Often however, the GEN algorithm will come up with rivals which destroy the relevant context for a particular constraint. Rivals which destroy this context will score better, since without the context, there can be no violation.

For an example, consider the evaluation of the personal name *hh.ba-ru-uk-*ka_4 against the proposed phonology /baruka/. When evaluating it for violations of the H2b constraint (the one which claims that the voicing of stops in the orthography represents a tense/lax distinction), the software comes up with 2 violations, one for *uk* against /k/ (i.e. the voiceless *k* corresponding to a lax /k/), and another for ka_4 against /k/. However, the GEN function for a different constraint, the one dealing with final vowels, produces a perfectly plausible rival

candidate *hh.ba-ru-uk*. When the H2b constraint counts the voiceless/tense violations on *hh.ba-ru-uk* vs. /baruka/, it only finds with one violation, for the *uk* grapheme. While the software may make this evaluation, it is obvious to the human observer that there is no real sense in which the *hh.ba-ru-uk* rival violates the voiceless/tense constraint more than *hh.ba-ru-uk-ka*$_4$ does. Nonetheless, the Gradual Learning Algorithm will go ahead and penalize the H2b constraint because it incorrectly ranked the rival *hh.ba-ru-uk* ahead of the selected exemplar form, *hh.ba-ru-uk-ka*$_4$.

The solution to this sort of problem would require the evaluation of constraints to be considerably more complex. In order to evaluate a particular rival orthography, the evaluation function would need to be aware of how many contexts were present where a violation could potentially occur, even if they were not present in the rival's actual orthography. Implementing such an approach in software would significantly complicate the constraint-evaluation code.

A much simpler fix to the problem would be to count the ratio of violations versus the number of contexts where the violation could occur, instead of simply counting the absolute number of violations. In such a case, *hh.ba-ru-uk* and *hh.ba-ru-uk-ka*$_4$ would be tied, with each of them scoring 100% violation of the H2b constraint. This would seem to be a more accurate description of reality than claiming that *hh.ba-ru-uk* violates the H2b (<voiceless>=/tense/) constraint less than *hh.ba-ru-uk-ka*$_4$ does.

However, Optimality Theory has traditionally proceeded by counting up the violations and cancelling off any matching marks. Treating violations as percentages rather than marks would represent something of a philosophical departure from mainstream Optimality Theory.

However, it should be noted that the strategy of counting violations and cancelling matching marks may well be just a reflection of the way that linguists are accustomed to creating tableaux by hand, rather than being a strategy which has an empirically-grounded basis. If that is so, it may be worth pursuing the

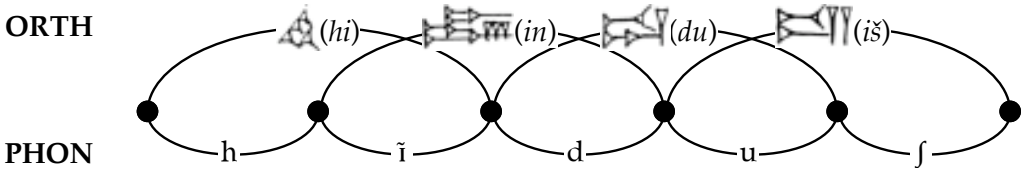
notion of measuring the ratio of violations per relevant context rather than just the absolute number of violations. In most cases, the two strategies will produce the same result, but there may be cases in fields other than orthographic analysis where a percentage measurement produces better results than the existing mark & cancel model.

8.2 Discussion of alignment

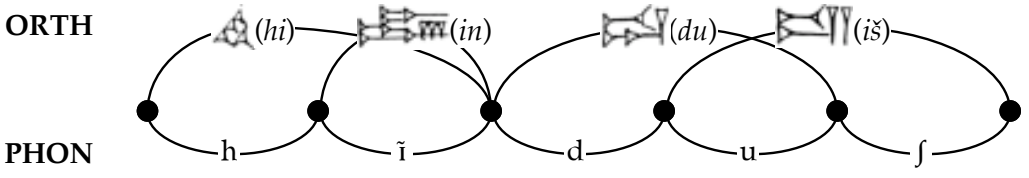
As discussed in §6.2, a great deal of work was put into developing an algorithm which could figure out which phonemes are supposed to be licensing which graphemes. Despite the 95% success rate on Old Persian loanwords, there were still numerous cases where the software had to disregard certain generated forms due to the inability to determine the licensing.

Even when the alignment algorithm has nominally succeeded, the algorithm can have some unexpected side-effects on the evaluation of the constraints. Consider the word for 'Indian', which shows up as *hi-du-iš, hi-in-du-iš,* or *in-du-iš*. It is not unreasonable to postulate an underlying phonology of /hĩduʃ/, based both on the range of written forms, and on the Old Persian phonology. However, when the alignment algorithm attempts to determine which phoneme sequences are licensing which graphemes, it has a difficult choice to make for the *in* grapheme. Licensing the vowel portion of *in* is straightforward, but what should be done for the consonant? If the software assumes that the important features are [+consonantal] and [-syllabic], it produces the first annotation graph shown in (23). However, if the software assumes that the important features are [+nasal] and [+sonorant], we get the second graph.

(23) Two annotation graphs for *hi-in-du-iš* 'Indian'



1 H4a (nasal vowel) violation
0 H4b (nasal consonant) violations



0 H4a (nasal vowel) violations
1 H4b (nasal consonant) violation

As can be seen, the choice of how to license the *in* grapheme makes a difference for how the H4a and H4b constraints are evaluated. Using the weightings given in (19), the software will line the *in* grapheme up with /ĩd/, because the "distance" between /n/ and /d/ is less than that between /n/ and /ĩ/. Hence, the alignment algorithm chooses the first of the two annotation graphs given in (23). This has the result of prejudicing the learning algorithm in favour of H4b instead of H4a. Ideally, the alignment algorithm should be neutral with respect to the various constraints

The licensing of the *in* sign in this example is one case of several where it appears that using phonological segments as the basis for licensing may be the wrong thing to do. Perhaps a better representation would be to think of the second portion of the *in* sign in *hi-in-du-uš* as being licensed by a [+nasal] feature, without attempting to tie the feature down to either the /i/ or the /d/ segment.

Although Sproat (2000) uses phonemes to describe licensing in shallow orthographies, there is no nothing in his theory which specifically requires the phonological level to be described in terms of phonemes. In fact, Sproat indicates that he is employing segments merely as a "shorthand" for a set of

overlapping gestures. We should really think of the $M_{ORL \rightarrow \Gamma}$ function as being a mapping from phonological features (or bundles thereof) to graphemes.

8.3 Discussion of specific constraints

The following section will discuss the various constraints introduced in §6.3, and consider whether the constraint rankings perform the task they were assigned. Each run of the software produces a log file of roughly 450 megabytes in size which gives a complete account of every single constraint evaluation during the processing of the Gradual Learning Algorithm. The following section distills the contents of this log file into a brief evaluation of how well each constraint performed.

8.3.1 H1) Interpretation of broken *CV_1-V_2C* writings

Hypothesis H1d, that the value of V_2 in a CV_1-V_2C sequence is ignored, wins by a considerable margin. This is not surprising, given that Old Persian names and loanwords comprise a large portion of the word inventory, and these words overwhelmingly have a simple vowel where the Elamite orthography has a CV_1-V_2C sequence.

Given the quantity of Old Persian data, it is no surprise that the diphthong hypothesis, H1b, does poorly. While Old Persian diphthongs and /V_1V_2/ sequences do sometimes correspond to broken-vowel orthographies, just as often they do not. For instance, Old Persian words with an /ai/ diphthong tends to be written in Elamite with a simple vowel, as in /arbaira/→*h.ar-be-ra* or /axʃainam/→*ak-še-na-um*. Or consider the diphthong in the Old Persian personal name *Gadauka*, which has spellings of *hh.ka$_4$-du-uk-ku*, *hh.ka$_4$-da-u-ka$_4$*, *hh.ka$_4$-du-ka$_4$*, *hh.ka$_4$-du-uk-ka$_4$*, and *hh.ka-tam$_5$-ka$_4$*, none of which employ a CV_1-V_2C sequence to indicate the diphthong.

The presence of the Old Persian data also has a tendency to skew the results against H1c (the intermediate vowel hypothesis). Hinz (1987) reconstructs a three-vowel system for Old Persian with distinctive lengths but no /e/ or /o/

vowels. Consequently, any forms which have Old Persian cognates will have only /a/, /i/, and /u/ vowels, and will tend to penalize H1c. Hence, possibilities like Hinz' suggestion of an /o/ vowel in /zomip/→*za-u-mi-ip* will be swamped by the mass of Old Persian data.

8.3.2 H2) Voicing of stops

Of the various voicing-related constraints, the only one which fares well is H2d, the hypothesis that sequences like *uk-ba* represent a combination of the voicing feature of one grapheme with the place of articulation by another. It seems likely that Hinz (1987) was correct in presenting this proposal.

The other two constraints, H2a (<Voicing>=/Voicing/) and H2b (<Voicing>=/Tense/) are both penalized heavily. This makes it probably that the remaining hypothesis, H2c, which did not have a corresponding constraint, is the correct one, and that voicing in the orthography does not correspond to a distinction in the phonology. This is in accord with most of the writers on Elamite, with the notable exception of Grillot-Susini (1988). It should be pointed out that the alternations which Grillot-Susini presented as evidence of contrastive voicing are all from earlier periods of Elamite, and not from Achæmenid Elamite. So the possibility is still open that contrastive voicing might have been a feature of Old Elamite or Middle Elamite.

8.3.3 H3) Geminate consonants

The results for this set of constraints strongly support the hypothesis (Reiner, 1969) that geminate orthographies are an attempt to indicate voicelessness. In particular, the opposing hypothesis (Grillot-Susini and Roche, 1988) that geminate orthographies represent geminate phonologies ended up being very heavily penalized.

What was surprising was that the hypothesis H3c (McAlpin, 1982) ranked so poorly. McAlpin's claim was that *l-l* and *r-r* geminates represent a separate (possibly retroflex) phoneme from the non-geminate orthographies. The

problem here seems to be a side-effect of the process for generating input candidates.

Consider the Akkadian name Nabû-kudurri-uṣur, which appears with a variety of spellings in Achæmenid Elamite texts. The *ur-ri* sequence which occurs in those spellings would appear to be an ideal context for evaluating McAlpin's hypothesis. However, when generating input candidates, the various anti-GEN functions create 238 permutations (mostly due to the permutations of voicing), but only four of those input candidates contain an /ɻ/ phoneme, with the rest having an /rr/ or an /r/. Since the anti-GEN function produces so few /ɻ/ input candidates for the *ur-ri* orthography, it is likely that the software will find an /r/ in the underlying phonology, and score a violation against McAlpin's constraint.

The prejudice against /l/ and /ɻ/ highlights the importance of having a fair and balanced anti-GEN function. The proposal discussed in §8.4 for cross-permuting the results of the constraint-specific anti-GEN functions would probably also improve the results for McAlpin's hypothesis.

8.3.4 H4) Nasal vowels

Unfortunately the effectiveness of the constraints for evaluating nasals was undermined by the alignment issues raised in §8.2. Although constraint H4b (NasalConsonants) is ranked significantly higher than H4a (NasalVowels), this may be merely a side-effect of the alignment algorithm.

8.3.5 H5) Word-final vowels

In this case, the clear winner is H5c, namely that word-final vowels in the orthography represent the underlying phonology. The other two constraints, the word-final cluster hypothesis H5a (Paper, 1955) and the word-final schwa hypothesis H5b (McAlpin, 1982) are among the lowest-ranked of all the constraints in the system. If a prothetic vowel were in fact being used to indicate word-final clusters, such use must be quite infrequent.

That being said, it is possible that the results of this comparison are skewed by the presence of large numbers of Old Persian words which have a known stable final vowel. Every one of those words would end up penalizing H5a and H5b, and rewarding H5c. Perhaps by restricting the context in which H5a, H5b, and H5c are evaluated, a fairer comparison might result in higher rankings for the final-cluster and final-schwa hypotheses. One possibility might be to restrict the comparison to native-Elamite words, but that is not feasible with the software as it is currently designed.

8.3.6 H6) Sibilants

The question of whether *ṣV* and *Vṣ* graphemes are representing an underlying /z/ or an underlying /tʃ/ remains open. The two constraints are effectively tied, which suggests that the constraints do not have enough information to prefer one underlying form over the other.

With respect to /ts/ affricate proposed by Khacikjan (1998), the problem is that it is difficult to judge whether the low ranking value of the H6c constraint is good or bad. In order to evaluate it, there really would need to be a countervailing constraint based on the hypothesis that there is no /ts/ affricate.

8.3.7 H7) The phonemic inventory includes an /h/.

Here again, the two constraints are effectively tied, which suggests that the constraints do not have enough information to prefer one hypothesis over the other.

8.3.8 H8) The phonemic inventory includes /f/ and /v/.

The hypothesis favouring the use of the *pír* grapheme to write /fr/ scores slightly lower than the hypothesis that it is being used to write /pr/ or /br/. On the whole though, both constraints rank fairly high. It seems likely that the *pír* grapheme was being used to write both /fr/ and /pr/. However, it seems that

there is not enough information to determine whether Elamite had an /f/ phoneme or not.

8.3.9 H9) The phonemic inventory includes a /j/

As was the case with the /h/ phoneme and the /z/ vs. /ʧ/ comparison, the constraint rankings neither support nor contradict the existence of a /j/ phoneme.

8.3.10 H10) The phonemic inventory includes a /w/

The hypothesis advanced by McAlpin (McAlpin, 1982) that *ú* is being used to write a separate /w/ phoneme seems to be unsupported. As pointed out by Khachikjan (1998), there are simply too many occurrences where *ú* is clearly being used to write a /u/ for McAlpin's hypothesis to be accurate. If there is a /w/ phoneme, it is not being written using the *ú* grapheme.

In fact, there is evidence from Old Elamite which suggests that the language did have a /w/ phoneme, but it was written using the *pi, ma,* or *me* graphemes, as in *ki-ir-pi-as, d.ki-ir-ma-áš, and d.kir-me-iš*, the name of an Elamite divinity whose name is rendered as Kirwas or Kirwaš. The use of the *pi* sign to write /wV/ is borrowed from Akkadian, but the use of the *ma* and *me* signs for /w/ would be uniquely Elamite. As Paper (1955) points out, the Elamite use of *ma, me, mi,* and *mu* to write /w/ continued into the Achæmenid period, when those signs are used to render Old Persian forms containing a /w/.

8.3.11 H11) The phonemic inventory includes an /e/

The constraint rankings weigh quite heavily in favour of there being an /e/ vowel in Elamite, distinct from /i/. This coincides with the position of everyone except for Paper (1955).

Unsurprisingly, the H11b constraint which evaluates the /e/ phoneme in word-initial syllables ranks higher than the H11a constraint which evaluates /e/ phonemes in general. This may significant, or it may merely reflect the fact that

there are naturally going to be more violations of H11a, since every violation of H11b automatically implies a violation of H11a as well.

8.4 Discussion of Lexicon Optimization

The generation of useful input candidates is limited by the information which is available to us. For all we know, Elamite had an /ɯ/ vowel, and *Grotefend* could even generate input candidates which contained an /ɯ/. However, none of the constraints would weigh either for or against it, so there is no point in generating such an input candidate. This does mean that the correct underlying form may well be inaccessible to our Lexicon Optimization technique. At best, Lexicon Optimization can produce an estimated underlying form which leaves as underspecified any features cannot be verified by a corresponding constraint. This is a limitation of Lexicon Optimization in general, not just of the implementation found in *Grotefend*.

Often, as we have seen in §8.3.6, §8.3.7, and §8.3.9, it appears that the constraint system lacks sufficient information to choose one underlying form over another. Although we cannot rule out the possibility that a more sophisticated set of constraints would be able to capture some distinction in such cases, it seems likely that the underlying forms are themselves underspecified. That is to say, we are unable to achieve our goal of determining the values for level II of our hierarchy because the values of level I are underspecified. To take the example of sibilants (§8.3.6), it may be impossible for the software to decide between /z/ and /tʃ/ simply because Elamite itself did not make such a distinction.

There is however, another limitation of our constraint-based generation of input candidates which should be fixed. By design, the anti-GEN functions work in isolation from each other. As can be seen from the detailed descriptions in §7, each of the anti-GEN functions produces input candidates based solely on what is found in the orthography.

Consider the operation of the anti-Gen functions on the orthography *da-iš*. The broken-vowel anti-GEN will produce /daiʃ/, /dajʃ/, /dɛʃ/, and /daʃ/.

Separately, the sibilant anti-GEN will produce /dais/, /daiʃ/, /daiz/, /daiʧ/, and /daiʦ/. Since the two functions operate independently, the software fails to generate a whole range of candidates such as /dajs/, /dajz/, /dajʧ/, /dajʦ/, /dɛs/, /dɛz/, /dɛʧ/, /dɛʦ/, /das/, /daz/, /daʧ/, and /daʦ/. If the actual underlying phonology happened to be /dɛʧ/, *Grotefend* would never find it, since that particular phonology will never be generated and presented to Lexicon Optimization as a possible input candidate.

A more sophisticated anti-GEN implementation would allow for the input candidates produced by one constraint's anti-GEN function to be further permuted by the anti-GEN function of another constraint. The original design decision was that each anti-GEN process would know only about the form's orthography, but this was clearly the wrong choice, since it results in many plausible input candidates being omitted from Lexicon Optimization.

Late in the software's development cycle, an attempt was made to retrofit this sort of cross-permutation onto the anti-GEN functions for some of the constraints. As a result, the anti-GEN functions for voicing (H2) and sibilants (H6) also generate some permutations for gemination (H3). However, in order to function effectively, the entire anti-GEN portion of the code should be redesigned and rewritten to generate the full range of input candidates.

9 Conclusions

This project represented an ambitious expedition into three largely unexplored territories: the application of Optimality Theory to orthography, the implementation of Lexicon Optimization in software, and the mass analysis of Elamite phonology. All three presented unanticipated challenges.

Given that there was essentially no prior work on applying Optimality Theory to orthography, the techniques introduced in this study represent an early attempt at working in this area. It is hoped that over time these techniques will evolve into something with a greater level of sophistication.

The problem of implementing GEN algorithmically appears to be at an early stage even in the processing of phonological data. The notion of constraint-driven GEN adopted from Heiberg (1999) does appear to be a useful starting point for working with orthography.

As can be seen from the case of the nasal vowel constraints, the determination of the mapping between phonology and orthography can have unexpected consequences for the evaluation of constraints. In part, the problem of alignment may have been made more difficult by the peculiarities of the cuneiform system. Other writing systems may present fewer difficulties in determining the $M_{ORL \rightarrow \Gamma}$ mapping in software.

Implementing meaningful constraints to evaluate the mismatches between phonology and orthography proved to be surprisingly complex. Part of the complexity may arise from the representation of the underlying phonologies as a sequence of phonemes. An alternative representation might be more effective, one which licensing graphemes based on bundles of features rather than by phonemes.

Connected to the implementation of constraints is the question of which contexts a constraint should be evaluated in. A powerful GEN function can easily create rivals which destroy the context for evaluating a particular constraint. This particular problem would seem to be an issue for Optimality Theory in general, not just for its application to orthography. The current position of Optimality Theory effectively is to assume that a destroyed context is a non-violation of the constraint.

The whole area of Lexicon Optimization has received surprisingly little mention in the literature of Optimality Theory. The notion that there must be some form of anti-GEN module to produce suitable input candidates appears never to have been raised at all. The existence of anti-GEN is hardly specific to the study of orthography, but would seem to be an omission from Optimality Theory in general.

The constraint-driven implementation of the anti-GEN function does seem like a promising strategy, although the details need work. In particular, there is a need for the outputs of the various constraint-specific anti-GENs to be permuted together in order to produce all plausible input candidates.

As for Elamite, the language has always been problematic both due to its status as an isolate and because the clues which are available end up being obscured by the writing system. The idiosyncracies of cuneiform meant that an inordinate amount of software development effort was devoted to the problem of aligning phonology and orthography. At most, we can claim that this application of software to analyze the body of Elamite vocabulary has only succeeded in duplicating some of the tentative conclusions drawn from a century of hard work "by hand".

Given the scope of the three unexplored territories that this study hoped to survey, it is hardly surprising that much is left to be done. The original goal of using Optimality Theory to reconstruct the phonology of an unknown language has proven to be over-ambitious. However, it is hoped that the tools and the insights developed along the way may prove to be useful to future explorers.

Bibliography

Bird, Steven, and Liberman, Mark. 1999. A Formal Framework for Linguistic Annotation. Philadelphia: Department of Computer and Information Science, University of Pennsylvania.

Boersma, Paul. 1997. How We Learn Variation, Optionality, and Probability. *Proceedings of the Institute of Phonetic Sciences of the University of Amsterdam* 21:43-58.

Boersma, Paul, and Hayes, Bruce. 2001. Empirical Tests of the Gradual Learning Algorithm. *Linguistic Inquiry* 32:45-86.

Caplice, Richard I., and Snell, Daniel C. 1988. *Introduction to Akkadian*: Studia Pohl. Series maior. ; 9. Rome: Biblical Institute Press.

Eisner, Jason. 1997. Efficient Generation in Primitive Optimality Theory. *Rutgers Optimality Archive* #206.

Ellison, Mark T. 1995. Phonological Derivation in Optimality Theory. *Rutgers Optimality Archive* #75.

Frank, Robert, and Satta, Giorgio. 1998. Optimality Theory and the Generative Complexity of Constraint Violability. *Rutgers Optimality Archive* #228.

Gragg, G. B. 1996. Other Languages. In *The World's Writing Systems*, eds. Peter T. Daniels and William Bright, 58-70. Oxford: Oxford University Press.

Grillot-Susini, Françoise, and Roche, Claude. 1988. *Eléments de grammaire élamite*: Etudes elamites. Paris: Editions Recherche sur les civilisations.

Hall, Daniel Currie. 2000. Infinity Limited: Constraining a Constraint-Based Theory, University of Toronto: Generals paper.

Hallock, Richard Treadwell. 1969. *Persepolis Fortification Tablets*: University of Chicago Oriental Institute publications ; v. 92. Chicago: University of Chicago Press.

Hammond, Michael. 1997. Parsing Syllables: Modelling OT Computationally. *Rutgers Optimality Archive* #222.

Heiberg, Andrea. 1999. Features in Optimality Theory: A computational model, University of Arizona: Doctoral dissertation.

Hinz, Walther, and Koch, Heidemarie. 1987. *Elamisches Wörterbuch*: Archäologische Mitteilungen aus Iran. Ergänzungsband ; 17. Berlin: D. Reimer.

Itô, Junko, Armin, Mester, and Padgett, Jaye. 1995. NC: Licensing and Underspecification in Optimality Theory. *Linguistic Inquiry* 26:571-613.

Karttunen, Lauri. 1998. The Proper Treatment of Optimality in Computational Phonology. *Rutgers Optimality Archive* #258.

Khachikjan, M. L. 1998. *The Elamite Language*: Documenta Asiana, v. 4. Roma: Consiglio Nazionale delle Ricerche Istituto per gli Studi Micenei ed Egeo-anatolici.

König, Friedrich Wilhelm. 1965. *Die elamischen Königsinschriften*: Beiheft (Archiv fur Orientforschung) ; 16. Graz: Im Selbstverlage des Herausgebers [E. Weidner].

McAlpin, David W. 1982. Proto-Elamo-Dravidian: The Evidence and its Implications. *Transactions of the American Philosophical Society* 71:1-155.

McCarthy, John J. 2002. *A thematic guide to optimality theory*: Research surveys in linguistics. Cambridge, UK ; New York, NY, USA: Cambridge University Press.

Paper, Herbert H. 1955. *The Phonology and Morphology of Royal Achæmenid Elamite*. Ann Arbor: University of Michigan Press.

Prince, Alan, and Smolensky, Paul. 1993. Optimality Theory. *Rutgers Optimality Archive* #537.

Reiner, Erica. 1969. The Elamite Language. *Handbuch der Orientalistik I/II/1/2/2*:54-118.

Sproat, Richard. 2000. *A Computational Theory of Writing Systems*. Cambridge: Cambridge University Press.

Steve, M.-J. 1992. *Syllabaire élamite: Histoire et paléographie*: Série II, Philologie. Neuchâtel: Civilisations du Proche-Orient.

Stolper, Matthew W. 2001. Review of Margaret Khachikjan, The Elamite Language [2001]. *Journal of Near Eastern Studies* 60:275-280.

Tesar, Bruce. 1995. Computational Optimality Theory, Dept. of Computer Science, University of Colorado: Ph.D. thesis.

Tesar, Bruce, and Smolensky, Paul. 2000. *Learnability in Optimality Theory*. Cambridge, Mass.: MIT Press.